

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## **IMAGES ARE BEST AVAILABLE COPY.**

As rescanning documents *will not* correct images,  
Please do not report the images to the  
Image Problem Mailbox.

**Parallel pseudo-random generator for emulating a serial pseudo-random generator and method for carrying out same.**

Patent Number: ☐ EP0397079, A3  
Publication date: 1990-11-14  
Inventor(s): ROGER GEORGE ANDRE CHARLES (FR); POWELL WILLIAM EDWARD (US);  
WEEBER WILLIAM BERNARD (US)  
Applicant(s): ALCATEL NV (NL)  
Requested Patent: ☐ JP3068022  
Application Number: EP19900108555 19900507  
Priority Number (s): US19890351175 19890512  
IPC Classification: H04L25/49  
EC Classification: H04L25/03E3  
Equivalents: AU5495690, AU629933, CA2016635, ☐ US5031129  
Cited patent(s): US4965881

**Abstract**

A parallel pseudo-random generator for emulating a serial pseudo-random generator that generates serial outputs such that the next serial output value is based upon an Exclusive OR combination of at least two preceding serial output values the maximum preceding serial output value defined as the Pth preceding serial output value, where P is an integer greater than one; comprising: A) at least P latches, each latch having an output having a logic value 1 or 0 and an input operable upon receipt of a clock signal, for receipt of data for controlling the next logic value on the latch output; B) at least P Exclusive OR gates, each having at least two inputs and one output, each Exclusive OR gate output connected to a corresponding input of one latch so as to define the next value of the latch output upon receipt of the next clock signal; and C) means for connecting each input of each Exclusive OR gate to one latch output so that the output of each Exclusive OR gate represents the corresponding next value of the latch to which This Exclusive Or gate output is connected.

Data supplied from the **esp@cenet** database - I2

## ⑫ 公開特許公報(A) 平3-68022

⑤ Int. Cl.<sup>5</sup>G 06 F 7/58  
H 03 K 3/84

識別記号

B  
A

庁内整理番号

7056-5B  
8626-5J

⑬ 公開 平成3年(1991)3月25日

審査請求 未請求 請求項の数 4 (全31頁)

⑭ 発明の名称 直列疑似乱数列生成機をエミュレートするための並列疑似乱数列生成機及びその実行方法

⑮ 特 願 平2-122861

⑯ 出 願 平2(1990)5月11日

優先権主張 ⑰ 1989年5月12日 ⑱ 米国(US) ⑲ 351,175

⑳ 発 明 者 ウィリアム・エドワード・ボーウエル アメリカ合衆国、ノース・カロライナ・27614、ローリ、トロッターズ・リッジ・ドライブ・201

㉑ 発 明 者 ウィリアム・バーナード・ウェバー アメリカ合衆国、ノース・カロライナ・27502、アベックス、プラニー・フランク・ロード・7917

㉒ 出 願 人 アルカテル・エヌ・ベー オランダ国、1077・イツクス・イツクス・アムステルダム、ストラビンスキーラン・341

㉓ 代 理 人 弁理士 川口 義雄 外2名

最終頁に続く

## 明 細 書

## 1. 発明の名称

直列疑似乱数列生成機をエミュレートするための並列疑似乱数列生成機及びその実行方法

## 2. 特許請求の範囲

(1) 次の直列出力値が少なくとも2つの先行の直列出力値の排他的ORの組合せに基づくような直列出力を生成する直列疑似乱数列生成機をエミュレートするための並列疑似乱数列生成機であって、最大の先行直列出力値がP番目(但しPは1よりも大きい整数である)の先行の直列出力値として定義されており、

A) 各々が、論理値1または0を有する出力と、前記出力において次の論理値を制御するためにデータを受信するようにクロック信号受信時に動作可能な入力とを有する少なくともP個のラッチと、

B) 各々が少なくとも2つの入力と1つの出力とを有する少なくともP個の排他的ORゲートであって、該排他的ORゲートの各々の出力が、次のクロック信号受信時に前記ラッチの出力の次の値を定義するように、1つのラッチの対応する入力に接続されている排他的ORゲートと、

C) 前記排他的ORゲートの各々の出力が、この排他的ORゲートの出力が接続されているラッチの対応する次の値を与えるように、前記排他的ORゲートの各々の入力を1つのラッチの出力に接続する手段

とを包含する並列疑似乱数列生成機、

(2) 前記直列疑似乱数列生成機を定義する直列排他的OR組合せが、(Pを7とすると)6番目及び7番目の先行の直列出力値に基づいてその次の出力値を決定する請求項1に記載の並列疑似乱数列生成機、

(3) 前記ラッチの数が8であり、該ラッチが対応する出力Q0からQ7を有しており、各々が対応するラッチの入力に接続されている出力を有する対応する排他的ORゲートEx0からEx7の入力が、

Ex0の入力がQ4及びQ6に接続され、

Ex1の入力がQ5及びQ7に接続され、

Ex2の入力がQ0及びQ1に接続され、

Ex3の入力がQ1及びQ2に接続され、

Ex4の入力がQ2及びQ3に接続され、

Ex5の入力がQ3及びQ4に接続され、

Ex6の入力がQ4及びQ5に接続され、

Ex7の入力がQ5及びQ6に接続される

ようにラッチの出力に接続されている請求項2に記載の並列疑似乱数列生成機。

(4) 前記直列疑似乱数列生成機を定数する直列排他的OR組合せが、6番目及び7番目の先行の直列出力を組合せるものであり、前記ラッチの数が16であり、前記ラッチは対応する出力Q0～Q15を

Ex14の入力がQ4及びQ5に接続され、

Ex15の入力がQ5及びQ6に接続される

ようにラッチの出力に接続されている請求項1に記載の並列疑似乱数列生成機。

### 3. 発明の詳細な説明

#### 技術分野

本発明は、直列疑似乱数列生成機(PRG)またはスクランブラ( scrambler)の出力を、直列PRGの遅延する直列出力を与える複数の出力を備えた並列実現態様によりエミュレートするための回路及びその方法に関する。本発明は特に、適正な刻時(clocking)を保証するため及びデータストリームの潜在的な機密保護のために、高速データストリームを直列PRGと組み合わせる通信において使用される。このような通信データは高速であるが故に、直列PRGを相補形金属酸化物シリコン(CMOS)回路を使用して実現することはできない。従って、回路のクロック速度がCMOS回路の動作周波数内で

有しており、該並列疑似乱数列生成機の幅が16に等しく、更に、各々が対応するラッチの入力に接続されている出力を有する対応する16個の排他的ORゲートEx0～Ex15の入力が、

Ex0の入力がQ8及びQ12に接続され、

Ex1の入力がQ9及びQ13に接続され、

Ex2の入力がQ10及びQ14に接続され、

Ex3の入力がQ11及びQ15に接続され、

Ex4の入力がQ0及びQ2に接続され、

Ex5の入力がQ1及びQ3に接続され、

Ex6の入力がQ2及びQ4に接続され、

Ex7の入力がQ3及びQ5に接続され、

Ex8の入力がQ4及びQ6に接続され、

Ex9の入力がQ5及びQ7に接続され、

Ex10の入力がQ0及びQ1に接続され、

Ex11の入力がQ1及びQ2に接続され、

Ex12の入力がQ2及びQ3に接続され、

Ex13の入力がQ3及びQ4に接続され、

あるように、直列PRGをエミュレートする必要がある。

#### 発明の背景

同期光網仕様(SONET)が採用されてから高速デジタル通信に対する標準が設けられた(参照: American National Standards Institute, Inc. "Digital Hierarchy Optical Interface Rates and Formats Specification" 標準 TI.105-1988)。典型的にはこのようなデジタル通信は、データストリームが多数の隔合った0と1とで構成されている場合に、そうしなければ生じ得るであろうクロック信号の損失の可能性を最小化するために、疑似乱数列の直列スクランブル信号をデータストリームと組み合わせる。しかしながら直列データストリームは1秒当たり155メガビットまたはそれ以上で動作し得るので、直列PRGは、製造がより安価であり且つ対応するエミッタ結合処理(ECL)またはヒ化ガリウム(GaAs)回路より低い電力で動作

する好ましいCMOS回路ではなくて、個別のECL回路、ECL適用業務特定集積回路(ECL ASIC)またはヒ化ガリウム回路のような高速製造技術を使用し実現される必要がある。ECL及びGaAs回路の付加製造コスト及び電力要求は、付加的な熱を散逸するためにプリント回路基板面積をより大きくする必要があり、またもやCMOS回路、特にCMOS適用業務特定集積回路(CMOS ASIC)が好ましくなる。

CMOS回路は典型的には50メガビットより速いクロック速度で動作できないので、直列疑似乱数列生成機のクロック周波数を効果的に小さくするためにはある技術を使用することが必要である。本発明は、かかる技術と、任意の直列PRGの生成多項式に対して、及び等価の直列シフトレジスタの長さよりも大きい任意のサイズの、直列PRGからの連続出力を与える並列出力ワードに対して動作可能な回路を提供する。

こうして、比較的成本が低く電力消費量がよ

並列PRGは、直列PRGを効果的にエミュレートする帰還経路を選択することにより任意の数の出力(任意の大きさの $N$ )に拡張することができる。帰還経路は直列生成多項式と並列PRG実現態様の出力の大きさとに基づいている。それぞれ $N$ が8及び16の本発明の2つの好ましい実施例においては、対応する数のD型フリップフロップ(FF)を、シミュレートされる直列PRGの次の $N$ 個の連続する値に対応する次の $N$ 個の出力の値を決定するのに必要な帰還を提供する排他的OR(XOR)ゲートと一緒に使用する。これら2つの実現態様は、直列疑似乱数列生成機をシミュレートするため最小数の排他的ORゲートに対して設定された最適化基準を使用することにより最適化される。

#### 発明の目的

本発明の目的は、 $N$ 個の並列出力が直列疑似乱数列生成機の $N$ 個の連続出力値をエミュレートするための、直列疑似乱数列生成機の出力をシミュ

り小さいCMOS回路を、直列PRGの出力をエミュレートする並列PRGを製造するために使用することができる。

#### 発明の概要

並列疑似乱数列生成機は、直列PRGの次の入力値が直列PRGの複数の先行の出力の排他的OR(XOR)の組合せに等しい帰還相成において次々と動作する直列疑似乱数列生成機をエミュレートするものと説明される。例えば通信において典型的なスクランブル多項式は $1+X^4+X^7$ である。この多項式は、直列PRGの次の入力値が、該生成機の5番目の先行の値と該生成機の7番目の先行の値とを排他的OR演算した出力に等しいことを意味する。生成機の7番目の値の出力も典型的には、スクランブルされるべきデータと排他的OR演算されている。

もし直列PRGがクロック速度 $f_s$ を有するならば、並列PRGはクロック速度 $f_p = f_s/H$ ( $H$ は並列PRGの出力の数である)を有する。

レートする並列疑似乱数スクランブラ回路とその方法とを提供する。

本発明の別の目的は、得られる並列クロック周波数が任意に低く設定され得、従ってCMOS製造技術を使用して並列PRGの実現態様を提供し得るように $N$ の値を任意の大きさにすることができる前記並列PRGを提供することである。

本発明の更に別の目的は、次の $N$ 個の出力を決定するために $N$ 個の出力から必要な帰還を提供するための排他的ORゲートと一緒にD型フリップフロップを組み込んだ前記並列PRGを提供することである。

本発明の更に別の目的は、任意の直列PRG生成多項式に対して実行可能な前記並列PRGを提供することである。

本発明の他の目的は以下に明らかとなるであろう。

本発明の特徴及び目的がより充分に理解される

ように、添付の図面を参照して以下に本発明を詳細に説明する。

#### 実施例

従来から通信情報をスクランブルするためには直列疑似乱数列生成機を使用する必要があった。第1図に示したように、典型的な直列疑似乱数列生成機20(直列PRG)にはシフトレジスタ22として構成された複数の段が、各段における値が次の段へと最後の段に到達するまで伝送されるように組み込まれている。最後の段における値は典型的には、通信データストリームの1つのビットと排他的OR(XOR)演算され、XOR演算の結果が通信業務において実際に伝送される。排他的OR演算は、もし両入力が論理値1または論理値0であるならば出力は論理値0であり、もし入力がそれぞれ論理値1と論理値0またはこの逆であるならば出力は論理値1であると定義される。排他的OR演算を表わす真理表を表1に示す。

る。

再度第1図を参照すると、直列疑似乱数列生成機の動作は典型的には多項式：

$$1 + X^n + \dots + X^p$$

で定義され得ることが判る。これは、「+」が排他的OR演算を意味する特性多項式として公知である(本明細書中では全てこのように使用するものとする)。

この特性多項式に関係する帰還方程式は、

$$X^0 + X^n + \dots + X^p = 0$$

として誘導される。上記式から、

$$X^0 + X^0 + X^n + \dots + X^p = 0 + X^0$$

となるが、一般に「+」が排他的OR演算子であると  $X + X = 0$  及び  $0 + X = X$  であるので、

$$X^n + \dots + X^p = X^0$$

となる。この方程式は、シフトレジスタの次の入力値が  $X^n + \dots + X^p$  であることを意味する。

例えば同期光網(SONET)規格(American National

表1

2つの入力に対する排他的ORゲートの真理表

$X_1$	$X_2$	$f$
0	0	0
0	1	1
1	0	1
1	1	0

$X_1$ 及び $X_2$ は入力であり、 $f$ は出力である。

(これは、繰上げのない2を法(モジュロ)とする和算に等価である。)

ほとんどの通信適用業務における疑似乱数列生成機の目的は、通信ビットストリームパターンとは無関係に、伝送される実態の情報がおよそ同じ数の1と0を含むことを保証することである。こうすると、例えばもし通信ビットストリームが長く連続する1または0のパターンを含むならばより困難になるであろう通信ビットストリームにおけるクロック情報の保守管理が容易となる。このようなスクランブルはデータ暗号化にも有効であ

Standards Institute(ANSI)規格 T1.105-1988

としても公知である)においてはこの多項式は  $1 + X^6 + X^7$  である。第1図から判るように、この多項式は、シフトレジスタ6における値がレジスタ7における値と排他的OR演算され、その結果がP段シフトレジスタの段1における次の値となることを意味する。表4は、7段シフトレジスタの各段に対する出発値が論理値1である場合の7つの段における値を示す。この出発値は典型的には「シード(seed)」と称される。SONET標準においては、シードは典型的には直列PRGに対して全て1である。これから判るように、段1に対して生成される値はシフトレジスタの段を連続的に移動する。前記したように、段7からの出力は直列通信ビットストリームと排他的OR演算するためにも使用される。

このような生成機が疑似乱数列生成機と称される理由は、生成されるビットストリームが同じ出

発シード及び同じ多項式に対して常に同じであるからである。

SONET多項式は段 6 及び 7 の排他的 OR 演算を使用した。勿論、直列シフトレジスタの異なる段を合わせて排他的 OR 演算する他の多項式を使用することもできる。実際、所望であれば 2 つ以上の段を排他的 OR 演算してもよい。

一般には最大長の多項式、即ち最大カウント(クロックサイクル)後にそれ自体を繰り返す多項式が使用される。最大長の多項式に対するカウントの最大数は、 $n$  次多項式において  $2^n - 1$  である。例えば、3 次の多項式では最大多項式は  $1 + X^2 + X^3$  であり、非最大多項式は  $1 + X^1 + X^2 + X^3$  である。表 2 及び 3 から判るように、最大長多項式は 7 つの出力の後に繰り返し、一方、非最大長多項式は 4 つの出力の後に繰り返す。

本発明は、最大であろうとなかろうと任意の直列多項式を用いて適用可能である。

メガビット/秒を超える速度では、相補形金属酸化シリコン(CMOS)集積回路の製造は非実用的になる。実際に、使用可能速度が約 75メガヘルツを超えるCMOSの製造は実質的に不可能である。結果としてSONET標準に使用されるもののような高い伝送速度(例えば155メガビット/秒)に対しては、このような直列疑似乱数列生成機を使用するのであれば、エミッタ結合論理(ECL)またはヒ化ガリウム(GaAs)技術を使用して製造することが必要となる。上記両技術はCMOS技術と比較して、典型的には製造がより難しく、より多くの熱を生成するので発生した熱を散逸するための集積回路素子を設置するためにより大きな面積のプリント回路基板を必要とし、1論理ゲート当たりのコストがより高くなるという深刻な欠点を有する。

本発明は、その値が直列疑似乱数列生成機の連続出力を与える複数の並列出力を有する並列疑似乱数列生成機を提供することにより、高速度疑似

表2

次数3の多項式

$$X^2 + X^3$$

$$(最大長 = 2^3 - 1 = 2^3 - 1 = 8 - 1 = 7)$$

直列段番号

(クロックサイクル) 1 2 3

1 0 1 1

2 0 0 1

3 1 0 0

4 0 1 0

5 1 0 1

6 1 1 0

7 1 1 1

8 0 1 1

9 等

表3

次数3の多項式

$$X^1 + X^2 + X^3$$

直列段番号

(クロックサイクル) 1 2 3

1 0 1 1

2 0 0 1

3 1 0 0

4 1 1 0

5 0 1 1

6 等

このような直列疑似乱数列生成機は、通信ビットストリームの伝送速度が約50メガビット/秒を越えると集積回路の実現態様に問題が生じる。50

ランダムビットパターンの生成に対する一般的な解決策を提供する。かかる並列疑似乱数列生成機24は任意の所望の数の並列出力を有することができ、第2図に示した実施例では8個の出力を有しており、第4図に示した実施例では16個の出力を有している。並列疑似乱数列生成機のサイズは、並列ワードのサイズがスクランブル多項式の次数に等しいかまたはそれより大きい限りは、特定の適用業務に最も適した任意の値に設定することができる。ディジタル集積回路を使用する場合には、出力の数は一般に2の倍数に等しい値を有し、例えば8個、16個等の出力を有する。

第2図に示した実施例においては、疑似乱数列生成機は、その出力(Q0~Q7)が、エミュレートされる直列疑似乱数列生成機の8個の連続出力値を与える8個のラッチ26を有する。ラッチ26はD型のフリップフロップとすることができる。直列疑似乱数列生成機の出力が直列段7である表4を参

照すると、この 7 番目の段は最初の 7 つの直列クロックサイクル(直列クロックサイクル 0 ~ 6)に対しては論理値 1 を有し、次のクロックサイクル(直列クロックサイクル 7)に対しては論理値 0 を有することが判る。従って 8 ビット並列 PRG の出力 Q7 ~ Q0 は、表 5 に示したように直列 PRG における段 7 の 8 個の連続出力値を与えることができる。即ち表 5 から、Q0 出力はこの直列 PRG 出力段 7 の 8 番目の直列出力を与え、Q1 は段 7 の 7 番目の直列出力を与える等、同様に Q7 までこの直列 PRG の段 7 の最初の直列出力を与える。このパターンが新たな並列出力の各々に対して繰り返される。

以下に記載するように、並列 PRG の次の 8 つの出力 Q7 ~ Q0 は値 00000100 を有する。これら Q7 ~ Q0 に対する値は、表 4 に表された段 7 の時間出力 8 ~ 15 を、並列クロックサイクル 1 に対する並列出力(表 5 参照)と比較することから判るように、直列段 7 の次の 8 個の直列出力を与える。従って並

列 PRG の最初(0 番目)のフレームは直列 PRG の段 7 の最初の 8 つの直列出力(直列クロックサイクル)に対応しており、並列 PRG のフレーム 1 は段 7 の次の 8 つの直列出力(直列クロックサイクル 8 ~ 15)に対応する等となる。最初の並列フレームは、それ自体が特定の出発シーケンスまたはシードを有する直列 PRG をエミュレートする上でその動作を開始するための、生成機への並列シード入力である。

表 4

1・X<sup>6</sup>・X<sup>7</sup> 生成多項式に対応する直列疑似乱数列発生機

時間 (等価の直列クロックサイクル)	直列段番号 1 2 3 4 5 6 7	
0	1 1 1 1 1 1 1	並列フレーム A0 (8 ビットの場合)
1	0 1 1 1 1 1 1	
2	0 0 1 1 1 1 1	
3	0 0 0 1 1 1 1	
4	0 0 0 0 1 1 1	
5	0 0 0 0 0 1 1	
6	0 0 0 0 0 0 1	
7	1 0 0 0 0 0 0	
8	0 1 0 0 0 0 0	並列フレーム A0 (16 ビットの場合)
9	0 0 1 0 0 0 0	
10	0 0 0 1 0 0 0	
11	0 0 0 0 1 0 0	
12	0 0 0 0 0 1 0	
13	1 0 0 0 0 0 1	
14	1 1 0 0 0 0 0	
15	0 1 1 0 0 0 0	
16	0 0 1 1 0 0 0	並列フレーム A2
17	0 0 0 1 1 0 0	
18	0 0 0 0 1 1 0	
19	1 0 0 0 0 1 1	
20	0 1 0 0 0 0 1	
21	1 0 1 0 0 0 0	
22	0 1 0 1 0 0 0	
23	0 0 1 0 1 0 0	
24	0 0 0 1 0 1 0	並列フレーム A1
25	1 0 0 0 1 0 1	
26	1 1 0 0 0 1 0	
27	1 1 1 0 0 0 1	
28	1 1 1 1 0 0 0	
29	0 1 1 1 1 0 0	
30	0 0 1 1 1 1 0	
31	1 0 0 1 1 1 1	

表 4 (続き)

1・X<sup>6</sup>・X<sup>7</sup> 生成多項式に対応する直列疑似乱数列発生機

時間 (等価の直列クロックサイクル)	直列段番号 1 2 3 4 5 6 7	
32	0 1 0 0 1 1 1	並列フレーム A4
33	0 0 1 0 0 1 1	
34	0 0 0 1 0 0 1	
35	1 0 0 0 1 0 0	
36	0 1 0 0 0 1 0	
37	1 0 1 0 0 0 1	
38	1 1 0 1 0 0 0	
39	0 1 1 0 1 0 0	並列フレーム A2 (16 ビットの場合)
40	0 0 1 1 0 1 0	
41	1 0 0 1 1 0 1	
42	1 1 0 0 1 1 0	
43	1 1 1 0 0 1 1	
44	0 1 1 1 0 0 1	
45	1 0 1 1 1 0 0	
46	0 1 0 1 1 1 0	並列フレーム A5
47	1 0 1 0 1 1 1	
48	0 1 0 1 0 1 1	
49	0 0 1 0 1 0 1	
50	1 0 0 1 0 1 0	
51	1 1 0 0 1 0 1	
52	1 1 1 0 0 1 0	
53	1 1 1 1 0 0 1	並列フレーム A6
54	1 1 1 1 1 0 0	
55	0 1 1 1 1 1 0	

表 5

!・X<sup>n</sup>-X'生成多項式に対応する直列PRGをエミュレートする並列疑似乱数列生成機(幅=8ビット)

並列 クロックサイクル	直列 クロックサイクル	並列出力 Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
0	0-7	0 1 1 1 1 1 1 1
1	8-15	0 0 1 0 0 0 0 0
2	16-23	0 0 0 1 1 0 0 0
3	24-31	1 0 0 0 1 0 1 0
4	32-39	0 0 1 0 0 1 1 1
5	40-47	1 0 0 1 1 0 1 0
6	48-55	0 0 1 0 1 0 1 1

$$F2 = Q0 + Q1$$

$$F3 = Q1 + Q2$$

$$F4 = Q2 + Q3$$

$$F5 = Q3 + Q4$$

$$F6 = Q4 + Q5$$

$$F7 = Q5 + Q6$$

第2図においては更に並列疑似乱数列生成機の出力Q0~Q7は、対応する数のデータストリーム出力排他的ORゲート30に次々と与えられることが判る。排他的ORゲート30においては、各排他的ORゲートへの第2の入力には、Q7の排他的OR出力ゲート30への入力直列データの最初のビットと排他的OR演算され、出力Q6は直列データの次のビットと排他的OR演算される等、Q0が直列データの8番目のビットと排他的OR演算されるように、直列データストリームの1つのビットが与えられる。従って出力線32における出力信号は、8ビットマルチプレクサ(図示なし)を使用することにより直

## 直列PRGをシミュレートするための並列実現態様

### 解析

第2図から判るように、ラッチ28に加えて並列PRGには更に複数の排他的ORゲート28が組み込まれており、排他的ORゲート28は、ラッチが次の出力を生成するようにラッチへ入力として与えるためにラッチの種々の出力を組合わせる。第3図は生成機を起動し(ANDゲート34)、並列シードをロードし(ORゲート36)、且つ並列クロック信号38を与えるための追加論理回路を示す第2図に対応する概略図である。

第2図から判るように、8つのフリップフロップの入力D0~D7には関数F0~F7に関係する値が与えられる。これらの関数は表5Aに示した方程式で定義される。

表 5A

$$F0 = Q4 + Q6$$

$$F1 = Q5 + Q7$$

列ビットストリームに変換し直され得るスクランブルされた出力データを与える。

第2図から、もし並列疑似乱数列生成機が幅8(W=8)を有するならば、シミュレートされた直列疑似乱数列生成機の次の8つの出力をそれぞれQ7~Q0で与えられるように各々の並列計算がなされるので、並列動作の周波数は到来する直列ビットストリームのものの8分の1となることは容易に判る。

### 並列出力排他的OR組合せの決定

以下により充分に説明するように、並列疑似乱数列生成機の各ラッチへの入力を与えるための排他的ORゲート構成は、直列疑似乱数列生成機の出力ビットストリームをエミュレートするように決定される。第2図には特定の排他的ORゲート構成を示してあるが、実際には多くの実現態様が可能である。本発明は、各入力に対して最小数の排他的ORゲートを使用する場合に特に有利である。こ

の構成は、直列ゲートに対する必要条件を最小化し、結果として各直列ゲートに付随するゲート遅延を最小化する。

任意の疑似乱数列生成機の多項式に対して、並列PRFの幅が、直列PRGの多項式を定義するのに使用される最大シフトレジスタ段に少なくとも等しいという条件で、排他的ORゲートを並列疑似乱数列生成機を実行するために使用することができる解決策が存在することは、経験的に見出されており、本明細においても発明者C. Rogerによる課題“並列疑似乱数列発生機、数学的解析(Parallel Psedo-Random Generator, Mathematical Analysis)”の数学的解析のなかに記載されているように数学的検証もされている。

第1図に関して与えられた上記多項式、即ち段1への次の入力段6及び7の排他的OR出力に等しい多項式においては、この関係は一般に、

$$Q(n) \equiv Q(n+8) + Q(n+7) \quad (1)$$

6参照))。即ち8ビット並列PRGの実現態様においては、Q7に対する次の値はQ-1に等しく、従ってQ5及びQ8の排他的ORに等しいことになる。即ち、次の $Q7 = F7 = Q5 + Q6$ である。

この同じ関係を用いてQ6～Q2の次の値は以下のように定義され得る。即ち、

$$\text{次の } Q8 = F8 = Q4 + Q5,$$

$$\text{次の } Q5 = F5 = Q3 + Q4,$$

$$\text{次の } Q4 = F4 = Q2 + Q3,$$

$$\text{次の } Q3 = F3 = Q1 + Q2,$$

$$\text{次の } Q2 = F2 = Q0 + Q1$$

となる。

Q1の数値の求め方は表6及び第7図を参照することにより最も良く理解され得る。

[式中nは直列PRGの任意の段である]

と定義され得ることが判る。第7図はこの関係を図で表したものである。

再度表4を参照すると、クロックサイクル0に対する段6及び7がともに論理値1を有することが判る。その結果、段1に対する次の値は0となる( $1+1=0$ (表1参照))。この結果は、nが0である場合の上記式と同等となる(次の直列クロックサイクル後には $Q(0)$ は $Q(1)$ となり、一般に次の直列クロックサイクル後には $Q(n-1)$ は $Q(n)$ となる)。

エミュレーション用8出力並列疑似乱数列生成機の次の8ビットを決定するためには、直列PRGの次に生成されるビットが、8直列クロックサイクル後に並列PRGの出力Q7に対する次の値となることが判る(Q-1は、1直列クロックサイクル後にQ0となり、更に7直列クロックサイクル後にQ7となり、これら8つの直列クロックサイクルは1つの並列ワードクロックサイクルと等価である(表

表6

2つの8ビットワードに対する並列出力値  
( $n=-8 \sim n=7$ )

Q-8	Q-7	Q-6	Q-5	Q-4	Q-3	Q-2	Q-1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
次の8ビットワード								現在の8ビットワード							

従って、(F1と等価の)Q1に対する次の値はQ-7の値に等しい。即ち、

$$\text{次の } Q1 = F1 = Q-7$$

である。式(1)から、

$$F1 = Q(-7+6) + Q(-7+7) = Q-1 + Q0 \quad (n=-7)$$

となるが、(式(1)を再度使用すると)、

$$Q-1 = Q5 + Q6 \quad (n=-1)$$

であるので、

$$\text{次の } Q1 = F1 = Q-1 + Q0 = Q5 + Q6 + Q0$$

と書ける。しかしながら更に式(1)から、Q0の現在の値はQ6+Q7の現在の値に等しい( $Q0 = Q6 + Q7$ )なので、

$$\text{次の } Q1 = F1 = Q5 + Q6 + Q6 + Q7 \quad (2)$$



$$\text{次の } Q5 = F5 = Q1 + Q3$$

$$\text{次の } Q4 = F4 = Q0 + Q2$$

である。

Q3の次の値はQ-13に等しい。上記式(1)を使用すると、

$$\text{次の } Q3 = F3 = Q-13$$

$$Q-13 = Q-7 + Q-8 \quad (n = -13)$$

$$Q-13 = (Q-1 + Q0) + (Q0 + Q1)$$

$$Q-13 = Q-1 + Q1$$

$$Q-13 = (Q5 + Q6) + Q1$$

$$Q-13 = (Q5 + Q6) + (Q7 + Q8) \quad (2a)$$

であり、更に式(1)から、

$$Q5 = Q11 + Q12$$

$$Q6 = Q12 + Q13$$

$$Q7 = Q13 + Q14$$

$$Q8 = Q14 + Q15$$

であり、従って、

$$Q-13 = (Q11 + Q12) + (Q12 + Q13) + (Q13 + Q14) + (Q14 + Q15)$$

任意の幅の並列PRGに対して使用することができ、直列多項式が、次の入力段を計算するために段6及び7を使用する上記実施例においては、Pの値は7であり、即ち並列PRGの幅は少なくとも7に等しい必要があるが、これより大きくともよい。

更に、直列多項式は2つの直列段の排他的ORに等しいとしたが、本発明は、次の入力段を計算するために排他的OR演算される直列段の数とは無関係に、任意の直列多項式に適用可能である。

より一般的な直列疑似乱数列生成機の例では多項式：

$$\text{次の直列入力} = X^2 + X^1 + X^0$$

を使用する。即ち特性多項式は $1 + X^2 + X^1 + X^0$ である。

この多項式は非最大(前記表2及び3参照)であるが、並列PRG実行方法が適用業務において一般的であること示すために与えられている。

となる。即ち、

$$Q3 = Q11 + Q15 \quad (2b)$$

である。同様に、

$$\text{次の } Q2 = F2 = Q10 + Q14$$

$$\text{次の } Q1 = F1 = Q9 + Q13$$

$$\text{次の } Q0 = F0 = Q8 + Q12$$

である。

上記解析に対応する16ビット並列PRGに対する排他的OR演算実現態様は第4図に示してある。

次のQ13は式(2a)及び(2b)によって示されるような複数の排他的OR演算によって定義され得ることが判る。一般に出力に対してはこのような多重表示が与えられ得る。1つの最適化基準は、出力Q3に対して式(2b)で示されるような最小数のゲート入力を使用することであろう。

上記解析は、並列PRGの幅がエミュレートされる直列PRGに対する繰返構成において使用される最大数の直列段に少なくとも等しいという条件で

第8図は、段nに関して、

$$Q(n) \equiv Q(n+2) + Q(n+5) + Q(n+9) \quad (3)$$

であるこの並列疑似乱数列生成機を示す。

表8は、この多項式の36のクロックサイクル(クロックサイクル0~35)に対応する直列疑似乱数列生成機を構成する9つの段に対する直列段の値を示す。

表 8  
直列多項式  $= 1 \cdot X^2 + X^5 + X^8$   
直列段番号

(クロックサイクル)	1	2	3	4	5	6	7	8	9
0	0	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1
2	0	1	0	1	1	1	1	1	1
3	1	0	1	0	1	1	1	1	1
4	0	1	0	1	0	1	1	1	1
5	0	0	1	0	1	0	1	1	1
6	0	0	0	1	0	1	0	1	1
7	1	0	0	0	1	0	1	0	1
8	0	1	0	0	0	1	0	1	0
9	1	0	1	0	0	0	1	0	1
10	1	1	0	1	0	0	0	1	0
11	1	1	1	0	1	0	0	0	1
12	1	1	1	1	0	1	0	0	0
13	1	1	1	1	1	0	1	0	0
14	0	1	1	1	1	1	0	1	0
15	0	0	1	1	1	1	1	0	1
16	0	0	0	1	1	1	1	1	0
17	1	0	0	0	1	1	1	1	1
18	0	1	0	0	0	1	1	1	1
19	0	0	1	0	0	0	1	1	1
20	1	0	0	1	0	0	0	1	1
21	1	1	0	0	1	0	0	0	1
22	1	1	1	0	0	1	0	0	0
23	1	1	1	1	0	0	1	0	0
24	1	1	1	1	1	0	0	1	0
25	0	1	1	1	1	1	0	0	1
26	1	0	1	1	1	1	1	0	0
27	1	1	0	1	1	1	1	1	0
28	0	1	1	0	1	1	1	1	1
29	1	0	1	1	0	1	1	1	1
30	1	1	0	1	1	0	1	1	1
31	1	1	1	0	1	1	0	1	1
32	1	1	1	1	0	1	1	0	1
33	0	1	1	1	1	0	1	1	0
34	0	0	1	1	1	1	0	1	1
35	0	0	0	1	1	1	1	0	1

式(3)の関係を使用することにより、第8図に示した直列疑似乱数列生成機をエミュレートする幅  $M=9$  を有する並列疑似乱数列生成機の次の出力  $Q0 \sim Q8$  に対する値は以下ようになる。即ち、

$$\text{次の } Q8 = F8 = Q-1 = Q1 + Q4 + Q8$$

$$\text{次の } Q7 = F7 = Q-2 = Q0 + Q3 + Q7$$

$$\text{次の } Q6 = F6 = Q-3 = Q-1 + Q2 + Q6$$

$$= Q1 + Q4 + Q8 + Q2 + Q6$$

$$= Q1 + Q2 + Q4 + Q6 + Q8$$

$$\text{次の } Q5 = F5 = Q-4 = Q-2 + Q1 + Q5$$

$$= Q0 + Q3 + Q7 + Q1 + Q5$$

$$= Q0 + Q1 + Q3 + Q5 + Q7$$

$$\text{次の } Q4 = F4 = Q-5 = Q-3 + Q0 + Q4$$

$$= Q-1 + Q2 + Q6 + Q0 + Q4$$

$$= Q1 + Q4 + Q8 + Q2 + Q6 + Q0 + Q4$$

$$= Q1 + Q8 + Q2 + Q6 + Q0$$

$$= Q0 + Q1 + Q2 + Q6 + Q8$$

$$\text{次の } Q3 = F3 = Q-6 = Q-4 + Q-1 + Q3$$

$$= (Q-2 + Q1 + Q5) + (Q1 + Q4 + Q8) + Q3$$

$$= (Q0 + Q3 + Q7) + Q1 + Q5 + Q1 + Q4 + Q8 + Q3$$

$$= Q0 + Q7 + Q5 + Q4 + Q8$$

$$= Q0 + Q4 + Q5 + Q7 + Q8$$

$$\text{次の } Q2 = F2 = Q-7 = Q-5 + Q-2 + Q2$$

$$= (Q-3 + Q0 + Q4) + (Q0 + Q3 + Q7) + Q2$$

$$= ((Q-1 + Q2 + Q6) + Q0 + Q4) + (Q0 + Q3 + Q7) + Q2$$

$$= ((Q1 + Q4 + Q8) + Q2 + Q6) + Q0 + Q4$$

$$+ (Q0 + Q3 + Q7) + Q2$$

$$= Q1 + Q8 + Q6 + Q3 + Q7$$

$$= Q1 + Q3 + Q6 + Q7 + Q8$$

$$\text{次の } Q1 = F1 = Q-8 = Q-6 + Q-3 + Q1$$

$$= (Q0 + Q7 + Q5 + Q4 + Q8)$$

$$+ (Q1 + Q4 + Q8 + Q2 + Q6) + Q1$$

$$= Q0 + Q7 + Q5 + Q2 + Q6$$

$$= Q0 + Q2 + Q5 + Q6 + Q7$$

$$\text{次の } Q0 = F0 = Q-9 = Q-7 + Q-4 + Q0$$

$$= (Q1 + Q8 + Q6 + Q3 + Q7)$$

表9は、直列クロックサイクル0～35に対応する4つの並列クロックサイクルに対する並列疑似乱数列生成機の出力値を示す。これらの出力は、最初の38の直列クロックサイクルに対する直列疑似乱数列生成機の出力段9に対応することが判る。

表9

$1 \cdot X^8 + X^5 + X^2$  多項式に対応する直列PRCをエミュレートする並列疑似乱数列生成機(幅=9ビット)

直列 クロックサイクル	並列 クロックサイクル	並列 Q0	並列 Q1	並列 Q2	並列 Q3	並列 Q4	並列 Q5	並列 Q6	並列 Q7	並列 Q8
0	0-8	0	1	1	1	1	1	1	1	1
1	9-17	1	0	1	0	0	0	1	0	1
2	18-26	0	1	0	0	0	1	1	1	1
3	27-35	1	1	0	1	1	1	1	1	0

以上の説明から、疑似乱数列生成機の幅が直列疑似乱数列生成機に使用されている段の数に少なくとも等しい限りは、並列疑似乱数列生成機が実現可能であることが判る。更に、並列PRCを実行するために必要な排他的ORゲートの最小数は、少なくとも並列PRCが直列PRCに等しい幅を有する場合には、対応する直列PRCに使用されている排他的ORゲートの数に必ずしも等しくないことが判る。

以下数学的解析によって、並列PRCの幅が少な

#### 並列疑似乱数列生成機

##### 数学的解析

座

並列疑似乱数列生成機について、シフトレジスタを用いて構築されている従来の直列PRC生成機を置き換えるための解析を行なう。並列及び直列の生成機をそれぞれ第5A図及び第5B図に示した概略図によって説明する。

従来の解決策においては、P段シフトレジスタの幾つかの段から発信された信号は一緒に排他的OR(XOR)ゲートによって加えられ、得られた信号はレジスタの入力に供給され、帰還を生成する。連続する信号値の間には方程式：

$$S_n = A_1 S_{n-1} + A_2 S_{n-2} + \dots + A_p S_{n-p} \quad (4)$$

(式中、「+」はXORまたは2を法とする加算に対して使用されており、 $A_1, \dots, A_i, \dots, A_p$ は、段iが接続されているならば1であり、そうでなければ0である)

くとも直列PRC多項式を実現するのに必要な段の数に等しい場合に、直列PRCの並列PRCエミュレーションが常に存在することを立証する。

が成立している。これは、係数が2を法とする整数の体(フィールド)、即ち' $F(0,1)$ 'に含まれる方程式である。

信号を「Z変換」すると、

$$S(Z) = A_1 Z^1 S(Z) + A_2 Z^2 S(Z) + \dots + A_p Z^p S(Z) \quad (5)$$

または

$$P(Z) \cdot S(Z) = 0 \quad (6)$$

$$\text{但し } P(Z) = Z^q + A_1 Z^1 + A_2 Z^2 + \dots + A_p Z^p \quad (7)$$

となる。方程式(5)と(6)とは等価である。

$P(Z)$ はSの特性多項式であり、Sの「生成式(generator)」と考えてもよい。

多項式 $P(Z)$ は「既約及び素」であり(係数が $F(0,1)$ に含まれる、より小さい次数の多項式の積ではない)、原始根 $Z^q + 1 = 0 (q = 2^p - 1)$ を有し、系によって生成される数列は周期 $2^p - 1$ の疑似乱数列生成式となる。

並列生成機は、その入力信号がXORゲート網によって計算される多重出力ラッチ(例えば複数の

フリップフロップ)で構成されており、またラッチの出力信号はこのXORゲート網に供給される。

#### 基本事項

以下の事項はデジタル信号処理方法を専門としない読み手に有効であろう。

1) 「2を法とする整数」の体はただ2つの元、即ち0と1とを含み、2種類の演算、即ち乗算(AND)と加算(排他的ORまたはXOR)とにおいて、

$$0 \times 0 = 0, 0 \times 1 = 1 \times 0, 1 \times 1 = 1, \text{ 及び}$$

$$0 + 0 = 0, 0 + 1 = 1 + 0 = 1, 1 + 1 = 0$$

が成立する。係数がこの体にある多項式は以下の特性を有する。即ち、

$P(Z) = Q(Z)$  は  $P(Z) + Q(Z) = 0$  と等値である、または  $(1+Z)^2 = 1+Z^2$  (何故ならば  $2=0$  である) である。

2) もし  $S_n = S(nT)$  ( $T$  は時間間隔である) ならば、

$$S(Z) = \sum S_n Z^n \quad (Z \text{ は遅延演算子である})$$

であるような「Z変換」を使用する。何故ならば、

$$Z S(Z) + \sum S_n Z^{n+1} = \sum S_{n+1} Z^n$$

多くとも  $2^p$  ワードを包含することは容易に判る。従って多くとも  $2^p - 1$  個のヌルでないワードが存在し得、配列の周期は多くとも  $2^p - 1$  である。この周期は、「既約及び素」の多項式と称される特定の多項式を用いて得られる。

5) 「a」を  $P(Z) = 0$  の根とすると、

$$P(a) = a^0 + A_1 a^1 + A_2 a^2 + \cdots + A_p a^p =$$

$$P^2(a) = a^0 + A_1 a^2 + A_2 a^4 + \cdots + A_p a^{2^p}$$

である。これは、 $A_i = A_i$  であり、

且つ上記式の  $p$  個の根は  $a, \dots, a^{2^p-1}$  であるので

$a^{2^p-1}$  は  $P(Z) = 0$  のもう1つの根であるが故である。

また  $a^{2^p-1} = 1$  であるので累乗指数  $2^p$  を有する次の根は「a」に等しい。

#### 並列生成機

第5A図の並列生成機の「Z方程式」は、

$$Z^0 = A_{0,0} Z^0 + A_{0,1} Z^{2^{p-1}} + \cdots + A_{0,p-1} Z^{2^{p-1}}$$

$$Z^1 = A_{1,0} Z^0 + A_{1,1} Z^{2^{p-1}} + \cdots + A_{1,p-1} Z^{2^{p-1}}$$

であり、これは  $S(t-T)$  のZ変換であるからである。

$$3) \quad S_n = A_1 S_{n-1} + A_2 S_{n-2} + \cdots + A_p S_{n-p} \quad (5)$$

の解を求める場合に、通常、

$$S_n = C a^n \quad (C \text{ は定数である})$$

と置くと方程式(5)は、

$$a^n = A_1 a^{n-1} + A_2 a^{n-2} + \cdots + A_p a^{n-p}$$

$$\text{または } a^0 + A_1 a + A_2 a^2 + \cdots + A_p a^p = 0$$

となり、「a」は  $P(Z) = 0$  の根であるべきであることが判る。この方程式には  $p$  個の根が存在する。

これらの根は一般に0または1で表わすことはできないが、式(5)の一般解はそれらの連続累乗の線形組合せとなる。即ち、

$$S_n = a_1^n + a_2^n + \cdots + a_p^n$$

となり、これは  $P(Z)$  の根の対称関数、即ちもしこれらの係数が2を法とする整数の体に含まれるならば、0または1である  $P(Z)$  の係数の関数である。

4)  $p$  ビットを含む直列PNS生成機のシフトレジスタは、ヌル配列を生成するワード  $0, 0, 0, \dots$  を含み

$$\dots\dots\dots$$

$$Z^1 = A_{1,0} Z^0 + A_{1,1} Z^{2^{p-1}} + \cdots + A_{1,p-1} Z^{2^{p-1}}$$

$$\dots\dots\dots$$

$$Z^{2^{p-1}} = A_{2^{p-1},0} Z^0 + A_{2^{p-1},1} Z^{2^{p-1}} + \cdots + A_{2^{p-1},p-1} Z^{2^{p-1}}$$

である。

(その要素が  $A_{i,j}$  である) 行列はラッチの2つの連続する状態  $n-1$  と  $n$  との間の遷移行列(transition matrix)である。これらの係数は、出力  $j$  が入力  $i$  に一般にXOR回路によって連結されているか否かに応じて1または0となる。例えば方程式  $i$  は、

$$S_{n-1} = (A_{i,0} S_{n-1,0}) \text{ XOR } (A_{i,1} S_{n-1,1}) \text{ XOR } (A_{i,2} S_{n-1,2}) \cdots \text{ XOR } (A_{i,p-1} S_{n-1,p-1})$$

に対応しており、方程式  $i$  は、

$$Z^i = \sum_{j=0}^{N-1} A_{i,j} Z^{2^j} \quad \text{または}$$

$$Z^0 = Z^{2^p-1} \sum_{j=0}^{N-1} A_{i,j} Z^j = Z^{2^p-1} R_i(Z)$$

と書くことができる。

$R_i$ は、その係数が遷移行列の行  $i$  の要素である多項式である。

$T_i(Z) = Z^i + Z^{i-1} R_i(Z)$  は、 $T_i(Z) \cdot S(Z) = 0$  であるものとなるべきである。 $P(Z) \cdot S(Z) = 0$  は既知であるので、 $T(Z)$  が  $P$  の倍数であるならば、例えば  $T(Z) = P(Z) \cdot Q(Z)$  であり、すると、

$$\begin{aligned} T(Z) \cdot S(Z) &= P(Z) \cdot Q(Z) \cdot S(Z) \\ &= Q(Z) [P(Z) \cdot S(Z)] = 0 \end{aligned}$$

となる。

(この結果は、 $S$  の連続する値が  $P(Z) = 0$  の根の累乗の組合せであることを考慮することにより得ることができ、これは、 $T(Z) = 0$  が少なくとも  $P(Z) = 0$  と同じ根を有することを示唆する)。

ここで、多項式：

$$P = A_0 + A_1 Z + A_2 Z^2 + A_3 Z^3$$

を考える(理解し易いように特定の例を用いるが、誘導方法は一般的である)。 $P(Z)$  によって生成される数列は、

数列であれば 0 となる。更に、もし  $S'_0, S'_{n-1}, S'_{n-2}, S'_{n-3}$  が  $S$  の連続する値であるならば、計算した和は 0 となり、 $S'_{n-2}$  は  $S$  の次のサンプルとなる。以上から、もし  $T(Z)$  が  $P(Z)$  の倍数であるならば、

- 1) 期待通り  $T(X) \cdot S(Z) = 0$  であり、且つ
- 2) 「良いシード」が与えられたならば  $T(Z)$  は数列  $S$  を生成すると結論される。この結果は、数列  $S$  の一連のサンプルを意味する(別のシードを用いても  $Q$  によって数列が生成されるであろう)。

$T(Z)$  では、(常に 1 である)その最初の係数とその  $N$  番目の最後の係数( $R_i$  の係数)のみがゼロでないとすることができる。従って、必要とされるシードはラッチ内に含まれるサンプルに制限される。出発時には、ラッチに  $P(Z)$  によって生成される数列の一部を負荷することが必要である。全ての多項式  $T_i(Z)$  はラッチの次のシリーズのビットのサンプルを生成し、各クロック時に対して図の左側

$$A_0 S_{n-3} + A_1 S_{n-2} + A_2 S_{n-1} + A_3 S_n = 0$$

$$A_0 S_{n-2} + A_1 S_{n-1} + A_2 S_n + A_3 S_{n+1} = 0$$

$$A_0 S_{n-1} + A_1 S_n + A_2 S_{n+1} + A_3 S_{n+2} = 0$$

$$\dots\dots\dots$$

である。 $Q(Z) = B_0 + B_1 Z + B_2 Z^2$  と置くと、

$$\begin{aligned} T(Z) &= P(Z) \cdot Q(Z) = A_0 B_0 + (A_0 B_1 + A_1 B_0) Z + \\ &\quad (A_0 B_2 + A_1 B_1 + A_2 B_0) Z^2 + (A_1 B_2 + A_2 B_1 + A_3 B_0) Z^3 \\ &\quad + (A_2 B_2 + A_3 B_1) Z^4 + A_3 B_2 Z^5. \end{aligned}$$

$$T(Z) = C_0 + C_1 Z + C_2 Z^2 + C_3 Z^3 + C_4 Z^4 + C_5 Z^5$$

となる。

計算すると  $S'$  は数列であり、

$$\begin{aligned} C_0 S'_{n-3} + C_1 S'_{n-2} + \dots + C_5 S'_n \\ = A_0 B_0 S'_{n-3} + (A_0 B_1 + A_1 B_0) S'_{n-2} + \dots \end{aligned}$$

となる。従って、

$$\begin{aligned} B_0 (A_0 S'_{n-3} + A_1 S'_{n-2} + A_2 S'_{n-1} + A_3 S'_{n-2}) + \\ B_1 (A_0 S'_{n-2} + A_1 S'_{n-1} + A_2 S'_{n-2} + A_3 S'_{n-1}) + \\ B_2 (A_0 S'_{n-1} + A_1 S'_{n-2} + A_2 S'_{n-1} + A_3 S'_n) \end{aligned}$$

であり、この式は、 $S'$  が  $P(Z)$  によって生成される

に位置する  $N$  ビットシリーズ(状態  $n$ )は右側に位置するシリーズ(状態  $n-1$ )となる等が言える。

第 6 図は多項式  $T_i(Z)$  の係数の図である。1 に等しい係数は「X」で表してあり、それ以外は 0 に等しい。遷移行列の要素である  $R_i$  の係数は平行四辺形の内側にあり、問題となるのは、

- 1)  $P(Z)$  の倍数、または  $P(Z)$  によって生成される数列の生成式(これは等価である)であり、
- 2) 最初の係数を除いて平行四辺形に包含される係数を有し、
- 3) 最も単純な実行態様を与えるために最小の 1 に最小の項を有する

多項式を見付けることであることが判る。

「Bezout の関係」(表 10 参照)は上記最初の 2 つの条件を満足する多項式を計算することができるが、第 3 の条件は常に満足するとは限らない。

「良い」多項式を見付ける極めて単純な方法は、行列の各行、即ち平行四辺形内に含まれる 2 つの

ゼロでない係数を有する多項式に対して「試算及び検証(try and see)」を行なうことである。これらの多項式は数列Sの生成式としてテストされ、最初に見付けられたものを使用し、可能であれば行列の次の行に対してもそれを再度使用する。2つの係数を用いた検索が失敗した場合には3つまたはそれ以上の係数に対して検索する等となる。

多項式は本来単純であるけれども、数10秒間の計算時間を要することがある(12次の多項式、 $N=32$ に対して20秒(D.E.C. VAX8600コンピュータ)を要するが、7次の多項式で $N=8$ または18に対してはほぼ即座に結果がでる)。ここで $N$ は $M$ 、即ち並列出力の数と等価である。

#### 他の誘導方法

##### a) 並列システムの特性方程式

かかる並列生成機は、その要素が0または1(1はXOR演算を意味する)である遷移行列によって連結されているラッチの2つの連続する状態によって

で置き換える。

方程式 $S_k(Z)=0$ は、遷移行列の全ての行 $k(1 \leq k \leq N)$ に対して満足されるべき特性方程式である。

##### b) 特性方程式の特徴

$P(Z)$ の $P$ 個の根の連続累乗は方程式(4)の解である。信号 $S_n$ はこれらの累乗の線形組合せであり、 $P(Z)$ の根の対称関数である。

従って直列シフトレジスタと同じ信号を生成するために、 $S_k(Z)$ の根は $P(Z)$ の根を包含する必要がある。

$$S_k(Z) = P(Z) Q_k(Z) \quad (12)$$

(式中、 $S_k$ 及び $P$ は $Z$ 項を有するので、 $Q_k(Z)$ は少なくともかかる項を有するべき多項式である。)

である。この逆は成立するであろうか。

もし式(11): $S_k(Z) = P(Z) Q_k(Z)$ であれば、 $S_k(Z)=0$ は、 $P(Z)$ の根に対してのみでなく $Q_k(Z)$ の根に対しても真となる。ここで寄生根(parasitic root)として公知の問題が生じるが、「良いシード」(即

ちえられ得る。そして、第2の状態の信号の各々は第1の状態の信号に式:

$$Z^i = \sum_j B_{ij} Z^{j-1} \quad (8)$$

( $0 \leq i \leq N-1$  且つ  $0 \leq j \leq N-1$ ,  $B_{ij}=0$  または 1) によって依存する。 $N$ はラッチ内に含まれるビット数であり、シフトレジスタによる解決策のごとき1直列クロックサイクル当たり1つのビットを与えるのではなくて、PRGの $N$ ビット( $N$ は前記のごとき $M$ と等価である)のシリーズは、各並列クロックサイクルに対して発信される。 $k=N-i$ と置くと式(8)は、

$$Z^{N-i} = Z^N \sum_j B_{kj} Z^{j-1} = Z^N R_k \quad (9)$$

(式中、 $R_k = \sum_j B_{kj} Z^{j-1}$ は、行列の $k$ 行(または $(N-i)$ 行)の係数を $B_{kj}$ とする $(N-1)$ 次の多項式である)と書ける。式(8)を、

$$Z^0 = 1 = Z^N R_k \quad (10)$$

または

$$S_k(Z) = 1 + Z^N R_k = 0 \quad (11)$$

ち良いPRGの一部)を選択することにより、かかる寄生根の導入を回避することができる。これは、 $S_n, S_{n-1}$ 等がラッチ内に包含され且つPRGの一部であるならば $S_{n+1}$ は同じPRGのビットであるし、もし $S_{n+1}, S_n, S_{n-1}, \dots$ がPRGの一部であるならば $S_{n+2}$ はPRGのビットである等、 $S_n$ の連続する値を考慮することにより証明され得る。

全ての多項式 $S_k(Z)$ は、 $Z^0$ から出発して $P(Z)$ の倍数であらねばならない。このような多項式は、 $Z^0$ と $Z^{N-1}$ との間にのみ他の項( $P_k$ の項)を有し、逆に、かかる多項式は並列発生機には都合が良い。 $k$ の同じ値に対して $S_k(Z)$ の幾つかの等価の式が存在し得る。

$S_k$ の異なる式が存在し得、例えば以下の2つの多項式:

$$S_{k1}(Z) = 1 + Z^N R_{k1} = P(Z) Q_{k1}(Z) \quad (13)$$

$$S_{k2}(Z) = 1 + Z^N R_{k2} = P(Z) Q_{k2}(Z) \quad (14)$$

は有効と言える。

唯一の条件は、 $P_{k1}$ 及び $P_{k2}$ の次数がいずれも $N$ より小さいことである。

上記2つの式を減算することにより、  

$$S_{k1} - S_{k2} = Z^* (R_{k1} - R_{k2}) = P(Z) (Q_{k1} - Q_{k2}) \quad (15)$$
を得る。

第1に、多項式 $(S_{k1} - S_{k2})$ は $P(Z)$ で整除され、 $S_{k1}$ 及び $S_{k2}$ は「 $P(Z)$ を法として合同」であると見なされる。これは、 $S_{k2}$ が $P(Z)$ を基準として $S_{k1}$ の項を置換することにより得られることを意味する。

例えばもし $P(Z) = 1 + Z^* + Z^*$ であれば、 $S_{k1}$ において、 $1 + Z^* + Z^* = 0$ であるならば $Z + Z^* + Z^* = 0$ があるので、 $Z$ を $Z^* + Z^*$ で置き換えることができる。

第2に、ゼロ根をもたない素数である $P(Z)$ は $Z^*$ で整除されず、 $(Q_{k1} - Q_{k2})$ は $Z^*$ で整除される。 $(R_{k1} - R_{k2})$ もまた $P(Z)$ の倍数であり、 $R_{k1}$ 及び $R_{k2}$ は $P(Z)$ を法として合同である。

従って、行列の同じ行に対して等価の特性多項式 $S_k(Z)$ を与える(次数が $N$ より小さい)「 $P(Z)$ を法

$$1 = Z^* R_k + P(Z) Q_k(Z) \quad (17)$$

BEZOUTの関係を認識する(後述の表10の「BEZOUTの関係」参照)。

$P(Z)$ 及び $Q(Z)$ を2つの多項式とすれば、それらの最大公約数(GCD)または最大公因数(HCF)は、  

$$HCF(P, Q) = A(Z)P(Z) + B(Z)Q(Z) \quad (18)$$
で表わされる。 $A$ 及び $B$ は(ユークリッドの互除法から誘導される)極めて単純なアルゴリズムを用いて見つけることができ、 $B$ の次数は $P$ の次数よりも小さい。

$Z^*$ 及び $P(Z)$ は公因数を持たないので( $P$ は既約である)、それらのHCFは1であり、式(17)はBEZOUTの関係である。

全ての値の $k$ に対して、その次数が $P(Z)$ の次数 $P$ よりも小さい多項式 $R_k$ を決定することができる。式(17)において $k=1$ をとると $1 = ZR + PQ$ となり且つ $PQ$ の次数は少なくとも $P$ に等しいので、 $R$ の次数は多くとも $p$ に等しく、唯一の可能性は $R$ の次数 $= p -$

として合同」の幾つかの多項式 $R_k(Z)$ が存在し得る。

第6図は幾つかの問題点を示す。

1) 考慮すべき2つの座標系が存在すること:

1つは多項式 $S_k$ に対するものであり、1つは多項式 $R_k$ に対するものである。これらは平行四辺形の内側または辺上になくはならず( $k=1 \sim k=8$ )、 $R_k$ の係数1が許容される位置を「/」で記した。

2) 選択した実施例においては $P(Z)$ の2つの重要な倍数、即ち、

$$1 + X^* + X^* \quad \text{及び} \quad 1 + Z^* + Z^*$$

が存在し、 $S_k$ の非定数項は、上記倍数の全てが許容範囲内にあるならばそのうちの1つの非定数項とすることができる。

3) 幾つかの多項式 $S_k$ は同一であり(例えば $S_1 \sim S_6$ )、対応する $R_k$ は項の平行移動のみが異なる。

c) Bezoutの関係

$$S_k(Z) = 1 + Z^* R_k = P(Z) Q_k(Z) \quad (16)$$

または

1であり、 $Q$ の次数は0である。即ち少なくとも1つの多項式 $R_k$ は $p$ 個の項を有しており、遷移行列は少なくとも $p$ 個の列を有する必要がある。

従って、

1)  $N$ は少なくとも $p$ に等しくなければならない( $N$ (ここでは $N$ と同じ)と $p$ との間に見られる関係説明参照)。

2)  $P$ より大きい $k$ または $P$ に等しい $N$ に対して、かかる問題に対する少なくとも1つの解がある。

この解は一般に、典型的には最小のXOR回路を求めるが故に最適ではない。しかし $N > p-1$ であるならば、更に次数が $N$ より小さいという条件でBEZOUTの関係によって与えられるものを用いて $P(Z)$ を法として合同な多項式 $R_k$ を探すことにより解を向上する方法が存在する。

d)「発見的方法(Heuristic Solution)」

発見的方法は、2つ、3つまたはそれ以上の非定

数項を有する $P(Z)$ の倍数を系統的に探索することからなる。行 $k$ に対して2つの係数の解が見付かったならば、それを $k+1, \dots$ に対して出来る限り使用する。3つ以上の係数が必要であるならば、それらを行 $k$ に対してのみ使用し、次の行はより少ない係数を許容することが望まれるので、2つの係数を再度用いて開始する。テストするためには $P(Z)$ で除算し、余りがゼロ多項式か調べる。これは、高い値の $p$ 及び $N$ に対しては計算時間が膨張することがあるが、いずれにせよ最速解を導くことができる(幾つかの解が存在する場合がある)。少なくとも現時点では最初に見付かった解を採用する。

$S_k$ 多項式をテストする別の方法は、多項式が所与の特性多項式によって生成された疑似乱数列を生成し得るか直接に検証することである。この方法はプログラム「GSPA-E」(表11及びサブルーチンPOLYANCOEFXのTEST部分参照)において使用されて

$P$ が「良い」多項式であることを確認するため、並列システムのテストのための良い「シード」を準備するため、及び並列システムをテストする基準を設けるために、 $P(Z)$ に対応するPRG配列(Seq1)を生成する。

2) 行列の要素を計算する。

3) 結果を印刷する。

行列の係数テーブル

$N$ が32以下である場合には行列の図

4) 検証する。

配列(Seq2)を生成し、Seq1と比較する。

上記プログラムと一緒にサブルーチンファイルを使用する。サブルーチンファイルは全て本発明の目的に必要な2を法とする代数における全ての演算を包含する。

テーブル12は、GPSA-Eプログラムの実行による端末リストサンプルである。

テーブル13は、SONET多項式 $(1+Z^4+Z^7)$ の8、

いる。

目的に応じて勿論他の方法も可能である。例えば、条件を満足し得る多項式 $S_k$ のテーブルを計算し、そのなかから選択して行列を構築することもより優れているであろう。

勿論、 $(1+Z^4+Z^7)^2$ のような倍数も良いことは明らかである。

#### まとめ

問題は、最小の係数を有する $P$ の倍数を認知し、そのなかから、その非定数項が $P_k$ 多項式の範囲内にあるものを選択することである。

#### e) プログラム

プログラムは4つの部分を含む。

1) 初期化と、データ、即ち、

多項式 $P(Z)$ の次数 $p$ と、

$A_0$ 及び $A_p$ (これらは常に1である)以外の $P$ の係数とラッチのビット数 $N$ と

の入力を行なう。更に、

16、24、32及び64ビット並列ワード幅に対する幾つかのプリントアウトを包含する。

#### f) 参考文献

Error Correcting Codes.

W.Wesley Peterson and E.J.Weldon (MIT Press)

Error Correction Coding for Digital Communications.

G.C.Clark and J.B.Cain(PLENUM)

Shift Register Sequences

Solomon W.Colomb(Holden-Day, Inc.)

Sequences Pseudo-Aléatoires

Georges C.Roger(Laboratoires de Marcoussis, Note Interne)

State Variables for Engineers(John Wiley & Sons, Inc.)

P.Derusso, R.Roy and C.Close, pp158-186

#### 表10

「Bezoutの関係」

2つの整数 $a$ 及び $b$ 、または2つの多項式の最大公因数(HCF)を求めたい場合、そのアルゴリズムはいずれに対しても同様である。

まず  $a$  を  $b$  で除算する。即ち、

$$a = Q_0 b + R_1 \quad 0 \leq R_1 \leq b$$

とする。 $a$  及び  $b$  は HCF で割り切れ、更に  $R_1$  も HCF で割り切れる。次に  $b$  を  $R_1$  で除算する等を繰り返す。

即ち、

$$b = Q_1 R_1 + R_2 \quad Q \leq R_2 \leq R_1$$

$$R_1 = Q_2 R_2 + R_3 \quad Q \leq R_3 \leq R_2$$

.....

$R$  は次第により小さくなり、従って、

$$R_{n-2} = Q_{n-1} R_{n-1} + R_n$$

--- ( $R_n$  は HCF である)

$$R_{n-1} = Q_n R_n + R_{n+1} \quad R_{n+1} = 0$$

---

$R_{n-1}$  は  $R_n$  で整除され、 $R_{n-2}$  も  $R_n$  で整除され、.....

$a$  及び  $b$  も  $R_n$  で整除される。

従って  $R_n$  は  $a$  及び  $b$  の HCF であり、これがユークリッドの互除法である。

次に連続する余りの数列：

$$1 = a(Z) A(Z) + b(Z) B(Z)$$

の関係が得られる。

$A_n$  の次数は積  $Q_1 \cdot Q_2 \cdots Q_{n-1}$  の次数に等しく、

$B_n$  の次数は積  $Q_0 \cdot Q_1 \cdots Q_{n-1}$  の次数に等しいこと

は明らかであるし、計算することもできる。更に、

$A$  を多項式  $A$  の次数を意味するとすると、

$$\cdot Q_0 = \cdot a - \cdot b$$

$$\cdot Q_2 = \cdot b - \cdot R_1$$

$$\cdot Q_2 = \cdot R_1 - \cdot R_2$$

.....

$$\cdot Q_{n-1} = \cdot R_{n-2} - \cdot R_{n-1}$$

$$\cdot Q_n = \cdot R_{n-1} - \cdot R_n$$

となり、従って  $\cdot(Q_0 \cdot Q_1 \cdot Q_2 \cdots Q_{n-1}) = \cdot a = \cdot R_{n-1}$

である。

$a$  と  $b$  とが互いに素であると仮定すると、 $R_n$  はそれらの HCF であり、 $\cdot R_n = 0$  及び  $\cdot R_{n-1}$  は少なくとも 1 である。従って  $\cdot B_n$  は  $\cdot a$  よりも小さく、同様の理由で  $\cdot A_n$  は  $\cdot b$  よりも小さい。

$$R_1 = a - Q_0 b = A_1 a + B_1 b \quad \text{但し } A_1 = 1 \text{ 及び } B_1 = -Q_0$$

$$R_2 = b - Q_1 R_1 = A_2 a + B_2 b \quad A_2 = -Q_1 A_1 \text{ 及び } B_2 = -Q_1 B_1$$

$$R_3 = R_1 - Q_2 R_2 = A_3 a + B_3 b$$

$$A_3 = A_1 - Q_2 A_2 \text{ 及び } B_3 = B_1 - Q_2 B_2$$

.....

$$R_n = A_n a + B_n b$$

$$\text{但し } A_n = A_{n-2} - Q_{n-1} A_{n-1} \text{ 及び } B_n = B_{n-2} - Q_{n-1} B_{n-1}$$

故に、 $A_n$  及び  $B_n$  は  $A_1$  及び  $B_1$  から得られる。 $A_1 = 1$ 、

$B_1 = 0$  ( $-1$  は添字である) 且つ  $A_0 = 0$  及び  $B_0 = 1$  とする

と、 $R_n$ 、 $A_n$  及び  $B_n$  を与えるアルゴリズムは、添字 1 から出発して単純に実行される。

もし  $R_n$  が HCF であるならば、「Bezout の関係」:

$$\text{HCF}(a, b) = A a + B b$$

が得られる。

$a$  と  $b$  とが互いに素であるならば、 $R_n = \text{HCF}(a, b)$

$= 1$  である。 $a$  及び  $b$  が多項式であるならば  $R_n$  は定

数であり、係数が  $F(0, 1)$  に含まれる場合には 1 に

等しい定数である。そして、

表 II

```

5  ' INPUT 0 TO INDICATE THE END OF THE OPERATION.',//
20  CONTINUE
    WRITE (UNIT2,7)
    FORMAT (' RANK OF A COEFFICIENT EQUAL TO 1 ? ', $)
    READ (UNIT1,8) I1
    FORMAT (1)
    IF (I1.EQ.0) GO TO 25  I MEANS THE END OF THE INPUT
    P2(I1)-1
    IAO (4 IN P2(I1), AI IN P2(I1),...
    GO TO 10
    INERT CORR. TO INPUT
    CONTINUE
    C ..... THE CHARACTERISTIC POLYNOMIAL HAS BEEN GIVEN. VERIFICATION:
    C
    9500  WRITE (UNIT2,9500)
    FORMAT (/)
    CALL ECRIPOL (UNIT3,'CHARACTERISTIC POLYNOMIAL',DEG,P2,25,0)
    C
    WRITE (UNIT2,9)
    FORMAT ('/' OR '(RETURN-YES, IF NOT, INPUT: N) ', $)
    REP=
    READ (UNIT3,1) REP
    FORMAT (A)
    IF (REP(1:1).EQ.'N') GO TO 10  I TO STARTING POINT
    IF (REP(1:1).EQ.'Y') GO TO 10  I TO STARTING POINT
    C ..... THE MISSING COEFFICIENTS WERE ENTERED. IS THE POLYNOMIAL CORRECT?
    C ..... WE VERIFY THAT POLYNOMIAL P2 GENERATES A MAXIMUM P.N.S.
    CALL SEQUENCE (UNIT2,DEG,P2,A,SEQ1,INDIC)  (COMPUTES THE P.N.S.
    IF (INDIC.EQ.1) THEN
    WRITE (6,19)
    FORMAT (/) THE P.N.S. IS NOT MAXIMUM (11)
    GO TO 10  I TO STARTING POINT
    END IF
    C ..... THE CHARACTERISTIC POLYNOMIAL IS GOOD .....
    9000  CONTINUE
    30  WRITE (UNIT2,13)
    13  FORMAT ('/', 2) NUMBER OF SIMULTANEOUS BITS TO BE ISSUED
    C (NIP-1) ' $)
    READ (UNIT3,10) N
    IF (N.LT.P) GO TO 10  I N MUST BE AT LEAST EQUAL TO P
    14  FORMAT (1)
    WRITE (UNIT2,9)
    REP=
    READ (UNIT3,1) REP
    IF (REP(1:1).EQ.'N') GO TO 10
    C ..... END OF PARAMETERS INPUT.
    C .....
    C
    WRITE (UNIT2,9500)
    WRITE (UNIT2,20)
    FORMAT (X,' .....')
    20  WRITE (UNIT2,205) P
    205  WRITE (UNIT2,205) P
    FORMAT (' DEGREE OF CHARACTERISTIC POLYNOMIAL 1 ',12,/)
    CALL ECRIPOL (UNIT4,'CHARACTERISTIC POLYNOMIAL',
    C,DEG,P2,25,0)
    CALL ECRIPOL (UNIT3,'CHARACTERISTIC POLYNOMIAL',
    C,DEG,P2,25,0)
    WRITE (UNIT2,17) N
    FORMAT (/) N
    17  FORMAT ('/', N) OF SIMULTANEOUS BITS: ',1,/)
    C ..... PARAMETERS ARE NOW SUMMARIZED ON THE SCREEN AND PUT IN THE FILE.
    C .....
    C ..... END OF INITIALISATIONS
    C

```

```

C ..... PROGRAM GPSA .....
C ..... COMPUTATION OF THE TRANSITION MATRIX OF A PARALLEL GENERATOR ....
C ..... OF PSEUDORANDOM SEQUENCE .....
C ..... (G.ROGER LABONATOIRES DE MARCOUSSIS- OCTOBER 9 1988) .....
C .....
C ..... DECLARATIONS
C
    CHARACTER*1 NOM,REP
    CHARACTER*4 NOM1
    CHARACTER*75 LIGNE
    INTEGER DEG
    PARAMETER (DEG=130,NBRN=64,NBRN-1000) INBITS FOUR DEGREES-12
    C .. POLYNOMIALS LIMITED TO DEGREE 128. PARAMETERS MAY BE ADJUSTED AS NEEDED.
    C .. MODITS IS AT LEAST TWICE THE NUMBER OF BITS OF THE P.N.S.
    C .. P IS THE DEGREE OF THE CHARACTERISTIC POLYNOMIAL GIVEN FOR THE SERIES
    C .. GENERATOR. P2 IS THE CHARACTERISTIC POLYNOMIAL ITSELF.
    INTEGER P,P1(DEG),P2(DEG),Q(DEG),R(DEG)
    INTEGER AN (NBRN,DEG)  I THE TRANSITION MATRIX
    INTEGER A1(DEG)
    INTEGER B1(DEG)
    INTEGER SEQ1 (NBRNITS),SEQ2(NBRNITS)
    C .. SEQ1 AND SEQ2 ARE TWO P.N.S.
    C .....
    C ..... INITIALISATIONS AND INPUT OF PARAMETERS .....
    C .....
    I1DEG=DEG
    UNIT1=6
    UNIT2=6
    UNIT3=5
    UNIT4=10
    OPEN (UNIT=10,FILE='GPSA.DAT',STATUS='NEW')
    WRITE (UNIT2,10)
    WRITE (UNIT2,10)
    301  C /, PARALLEL PSEUDORANDOM SEQUENCES GENERATOR. '//,
    C /, COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM GPSA) '//,
    C /, GEORGES ROGER I.D.M. 9/11/88'///
    C .....
    C ..... INPUT OF PARAMETERS .....
    C .....
    10  WRITE (UNIT2,1)
    FORMAT ('/', 1) DEGREE OF THE CHARACTERISTIC POLYNOMIAL
    C (1(P(1)) ' $)
    READ (UNIT3,2) P
    FORMAT (1)
    IF (P.LT.2) THEN
    WRITE (UNIT2,1)
    FORMAT (' DEGREE TOO SMALL')
    GO TO 10
    END IF
    IF (P.GT.12) THEN
    WRITE (UNIT2,4)
    FORMAT (' DEGREE TOO BIG')
    GO TO 10
    END IF
    201  WRITE (UNIT2,20) P
    FORMAT (' DEGREE OF CHARACTERISTIC POLYNOMIAL 1 ',12)
    CALL PHUL (DEG,P2)
    C ..... CREATES THE CHARACTER. POLYN. WITH
    C ..... LAST AND
    C ..... FIRST COEFFICIENTS (ALWAYS EQUAL TO 1)
    C ..... THEN, ASKS FOR OTHER COEFFICIENTS
    C ..... INPUT OF THE CHARACTERISTIC POLYNOMIAL. '//,
    C /, POLYNOMIALS ARE WRITTEN AS: '//,
    C /, X0 + A1 X1 + A2 X2 + ... + AP-1 X P-1 + AP'//,
    C /, PLEASE GIVE THE RANK OF COEFFICIENTS A1 TO AP-1 EQUAL TO 1,
    C ..... AFTER THE OTHER. '//,
    C /, THOSE OF DEGREE 0 AND P ARE EQUAL TO 1 (ALREADY) '//,

```

```

C .....
C .....
207 WRITE (UNIT1,207)
    FORMAT (/,X, ' COMPUTING .....')
    WRITE (UNIT2,207)
C ** LOOKING FOR THE SOLUTION .....
2000 CONTINUE
    IF (KR.GT.N) GO TO 2300 ! THE END
    IDEP=KR
    NBC=2
    ! NO OF COEFF DE EX KR AUTRES QUE 10
    C ** WE LOOK FOR A POLYNOMIAL P1 MULTIPLE OF P1 AND HAVING ONLY
    C ** NBC-TWO COEFFICIENTS BETWEEN KR AND KR+1-1. SUCH A POLYNOMIAL IS A
    C2 GENERATOR OF THE P.N. 2. SEQ1, ALREADY COMPUTED.
    2001 WRITE (UNIT1,2001) KR, NBC-1, NBC
    FORMAT (/, KR, '1', ' NBC-1= ', '1', ' NBC= ', '1')
    CALL POLYANCOEFF (DEG.P1, KR, NBC-1, NBC, P1, INDICP, SEQ1)
    C3 IF (INDICP.EQ.1) WRITE (UNIT1,2003) INDICP ! FOR TEST OR DEBUGGING
    2003 FORMAT (/, INDICP= '1')
    C .....
    IF (INDICP.EQ.1) THEN ! SUCCESS
        P1(1)=0
        ! TO LOOK FOR THE DEGREE OF THE 1ST COEF OTHER THAN
        C ! 10, ALWAYS EQUAL TO 1
        C3 CALL DEGREE (P1, DEG, LOGGED, INDIC) ! LOGGED-DEGREE OF THE FIRST COEFF
        2005 WRITE (UNIT1,2005) LOGGED
        C .....
        C ..... P1 MAY BE USED UNTIL A-LOGGED .....
        C ..... POLYNOMIAL P1 IS COPIED IN THE MATRIX RH WITH THE CORRECT SHIFT
        DO 2100 KR=KR, LOGGED
            IF (KR.GT.N) GO TO 2900 ! FINISHED
            DO 2200 LL=1, N ! P1 COPIED IN RH
                RH (N-KR+1, LL)=P1(LL+KR)
            CONTINUE
        2200 CONTINUE
        NONH=RH
        WRITE (NONH(1,1), '(12,2)'//NONH(1,1))
        CALL ECRIPOLTAB (LOGED, NONH, UNIT4, NONH, DEG, RH, N-KR+1, 70, 0)
        2100 CONTINUE
        KR=LOGGED+1
        GO TO 2000
    END IF
    ! IF INDICP=0, NO SUCCESS, WE TRY WITH 3, 4, ETC. COEFF.
    C .....
    2300 CONTINUE
    NBC=NBC+1
    C3 WRITE (UNIT1,2001) KR, NBC-1, NBC
    C3 CALL POLYANCOEFF (DEG.P1, KR, NBC-1, NBC, P1, INDICP, SEQ1)
    IF (INDICP.EQ.1) WRITE (UNIT1,2003) INDICP
    IF (INDICP.EQ.1) THEN ! SUCCESS
        DO 2250 LL=1, N
            RH (N-KR+1, LL)=P1(LL+KR)
        CONTINUE
        NONH=RH
        ! P1 PUT IN RH
        WRITE (NONH(1,1), '(12,2)'//NONH(1,1))
        CALL ECRIPOLTAB (LOGED, NONH, UNIT4, NONH, DEG, RH, N-KR+1, 70, 0).
        RE=KR+1
        GO TO 2000
    END IF
    ! NEXT LINE OF THE MATRIX
    ! OR NO SUCCESS
    GO TO 2300
    ! ONE MORE COEFF.
    C .....
    2900 CONTINUE
    3000 CONTINUE
    C ..... THE END .....
    C ..... END OF THE COMPUTATION .....
    C ..... WRITING THE MATRIX IN THE FILE .....
    WRITE (UNIT4,601) (RH (1,2)) ! FORM FEED

```

```

IF (N.GT.22) THEN
    WRITE (UNIT4,611)
    FORMAT (/,/, ' MATRIX TOO BIG. UNABLE TO WRITE IT... ', //)
    GO TO 600
END IF
607 FORMAT (X,A)
    WRITE (UNIT4,9500)
    WRITE (UNIT4,203)
    WRITE (UNIT4,205) P
    CALL ECRIPOL (UNIT4, ' CHARACTERISTIC POLYNOMIAL ',
    DEG.P1, 75, 0)
    WRITE (UNIT4,17) M
    WRITE (UNIT4,203)
    WRITE (UNIT4,9500)
    WRITE (UNIT4,601)
    601 FORMAT (/,/, 10X, ' MATRIX(,/,)
    IF (N.LE.24) KR=3
    IF (N.GT.24) KR=1
    WRITE (UNIT4,605) ((JJ-1)/10, JJ-1, N)
    605 FORMAT (X, 40) ((KR-1), 3)
    WRITE (UNIT4,603) ((JJ-1)/10, JJ-1, N)
    603 FORMAT (X, 40) ((KR-1), 3)
    DO 600 II=1, N
        LIGNE=II
        DO 610 JJ=1, N
            JJ1=JJ-1
            IF (RH(11, JJ).EQ.0) LIGNE (JJ1, JJ1)=-1
            IF (RH(11, JJ).EQ.1) LIGNE (JJ1, JJ1)=-2
        CONTINUE
        610 WRITE (UNIT4,617) II-1, LIGNE
        617 FORMAT (X, 12, 2X, A25)
        CONTINUE
        600 WRITE (UNIT4,9500)
    C ..... THE MATRIX IS WRITTEN
    C .....
    C ..... VERIFICATION .....
    ! PERMANENT=PERMANENT
    NONH=DEGREE-1-DEGREE+100
    C ..... VERIFICATION OF THE PARALLEL GENERATOR .....
    C ** WE USE POLYNOMIAL 'A' FOR STATE 'N' AND 'B' FOR STATE 'M1'
    C ** WE LOAD THE GENERATOR WITH THE FIRST N VALUES OF SEQ1
    DO 5000 II=1, N
        A(N-II+1)=SEQ1(II)
        ! TAKE CARE OF THE TIME INCREASING DIRECTION!
        C ** WE NOW COMPUTE THE VALUES TO PUT INTO B1
        C .....
        DO 5500 M=1, NONHDEGREES/N
            DO 5100 II=1, N
                B(II)=0
                DO 5200 JJ=1, N
                    IF (RH(11, JJ).EQ.1) B(II)=B(II)+RH(1, JJ)
                CONTINUE
            CONTINUE
            ! WE PUT THE VALUES OF B IN SEQ2!
            DO 5300 II=1, N
                SEQ2(M-II+1)=B(II)-A(N-II+1)
            CONTINUE
        5300 CONTINUE
        ! WE PUT THE VALUES OF B IN 'A' FOR THE FOLLOWING STATE!
        DO 5400 II=1, N
            A(II)=B(II)
        CONTINUE
        5400 CONTINUE
        C .....
        C ** GENERATOR, AND COMPARISON WITH SEQ1, OBTAINED WITH THE PARALLEL
        WRITE (UNIT4,470) ((SEQ1(KR), KR-1, 72)
        470 FORMAT (/, ' SEQ1', ' ', 72(1)
        WRITE (UNIT4,570) ((SEQ2(KR), KR-1, 72)
        570 FORMAT (/, ' SEQ2', ' ', 72(1)

```



```

C
1  FORMAT (X,'DEGRE=1, POLYHOME (VENTIQUEUNUIT NULLI')
C
C  GO TO 10000
END IF
210  DEGR=11-1
10000  CONTINUE
      RETURN
END
C .....
C .....
SUBROUTINE COPIEX (DEG,P1,P2)
C ** RECOPIE P1 DANS P2 TERME A TERME, COPIES P1 INTO P2
INTEGER DEG
INTEGER P1(1),P2(1)
DO 100 I1=1,DEG
P2(I1)=P1(I1)
100  CONTINUE
      RETURN
END
C .....
C .....
SUBROUTINE PHUL (DEG,P)
C ** CREER LE POLYHOME NUL DE DEGRE MAX DEG. CREATES THE NULL POLYNOMIAL
INTEGER DEG
INTEGER P(1)
DO 100 I1=1,DEG
P(I1)=0
100  RETURN
END
C .....
C .....
SUBROUTINE PUNIT (DEG,P)
C ..... CREER LE POLYHOME DE DEGRE 0, CREATES THE CONSTANT POLYN. ....
INTEGER DEG
INTEGER P(1)
CALL PHUL (DEG,P)
P(1)=1
      RETURN
END
C .....
C .....
SUBROUTINE ECRIPOL (UNIT,NOM,DEG,P,DEGMAX,INDIC)
C ** INDIC=0, TERMES BLOQUES, SI INDIC=1, ESPACES DE 4 BLANCS POUR CHAQUE
C ** TERME NUL
C ** UNIT=1, UNITE LOGIQUE SUR LAQUELLE ON ECRIT.
C ** UNIT=2, UNITE LOGIQUE SUR LAQUELLE ON ECRIT.
C ** UNIT=3, UNITE LOGIQUE SUR LAQUELLE ON ECRIT.
C ** FOUR BLANKS ARE LEFT FOR EACH BLANK TERM. DEGMAX LIMITS THE NUMBER OF
C ** WRITTEN TERMS. NOM IS THE NAME OF THE POLYNOMIAL, WHICH MAY BE WRITTEN.
CHARACTER*(*)NOM
CHARACTER*40 LICNE
INTEGER DEG,DEGMAX,DEGMAX1
INTEGER P(1)
CALL DEGRS (P,DEG,DEGMAX1,INDIC?)
C=LEN (NOM)
LICNE=' '
LICNE (1:1)=NOM
LICNE (C:1+C)= ' '
100P=L+1
C .....
C .....
DO 100 I1=1,DEGMAX
IF (P(I1).EQ.0.AND.INDIC.EQ.1) GO TO 120
IF (P(I1).EQ.1) THEN 11
LICNE(100P:I00P)=I1
IF (I1=1).LE.9 THEN LICNE(100P+1:100P+1)=I1
IF (I1=1).GT.9 AND (I1=1).LT.100
C WRITE(LICNE(100P+1:100P+1),'(I1)')I1-1
120  CONTINUE
100  CONTINUE
END

```

```

10 IF (I(1)-1).GT.99.AND.(I(1)-1).LT.1000)
110 C WRITE(LIGNE(I(1),I(1)), '(I(1))I(1)-1
120 I(1)-I(1)+1'
130 I(1)=I(1)+1
140 I(1)=I(1)+1
150 I(1)=I(1)+1
160 I(1)=I(1)+1
170 I(1)=I(1)+1
180 I(1)=I(1)+1
190 I(1)=I(1)+1
200 I(1)=I(1)+1
210 I(1)=I(1)+1
220 I(1)=I(1)+1
230 I(1)=I(1)+1
240 I(1)=I(1)+1
250 I(1)=I(1)+1
260 I(1)=I(1)+1
270 I(1)=I(1)+1
280 I(1)=I(1)+1
290 I(1)=I(1)+1
300 I(1)=I(1)+1
310 I(1)=I(1)+1
320 I(1)=I(1)+1
330 I(1)=I(1)+1
340 I(1)=I(1)+1
350 I(1)=I(1)+1
360 I(1)=I(1)+1
370 I(1)=I(1)+1
380 I(1)=I(1)+1
390 I(1)=I(1)+1
400 I(1)=I(1)+1
410 I(1)=I(1)+1
420 I(1)=I(1)+1
430 I(1)=I(1)+1
440 I(1)=I(1)+1
450 I(1)=I(1)+1
460 I(1)=I(1)+1
470 I(1)=I(1)+1
480 I(1)=I(1)+1
490 I(1)=I(1)+1
500 I(1)=I(1)+1
510 I(1)=I(1)+1
520 I(1)=I(1)+1
530 I(1)=I(1)+1
540 I(1)=I(1)+1
550 I(1)=I(1)+1
560 I(1)=I(1)+1
570 I(1)=I(1)+1
580 I(1)=I(1)+1
590 I(1)=I(1)+1
600 I(1)=I(1)+1
610 I(1)=I(1)+1
620 I(1)=I(1)+1
630 I(1)=I(1)+1
640 I(1)=I(1)+1
650 I(1)=I(1)+1
660 I(1)=I(1)+1
670 I(1)=I(1)+1
680 I(1)=I(1)+1
690 I(1)=I(1)+1
700 I(1)=I(1)+1
710 I(1)=I(1)+1
720 I(1)=I(1)+1
730 I(1)=I(1)+1
740 I(1)=I(1)+1
750 I(1)=I(1)+1
760 I(1)=I(1)+1
770 I(1)=I(1)+1
780 I(1)=I(1)+1
790 I(1)=I(1)+1
800 I(1)=I(1)+1
810 I(1)=I(1)+1
820 I(1)=I(1)+1
830 I(1)=I(1)+1
840 I(1)=I(1)+1
850 I(1)=I(1)+1
860 I(1)=I(1)+1
870 I(1)=I(1)+1
880 I(1)=I(1)+1
890 I(1)=I(1)+1
900 I(1)=I(1)+1
910 I(1)=I(1)+1
920 I(1)=I(1)+1
930 I(1)=I(1)+1
940 I(1)=I(1)+1
950 I(1)=I(1)+1
960 I(1)=I(1)+1
970 I(1)=I(1)+1
980 I(1)=I(1)+1
990 I(1)=I(1)+1
1000 I(1)=I(1)+1

```

致12

SAMPLE TERMINAL LISTING  
- -  
RUNNING PCH GP5A - E

\$ RUN CP5A E

# PARALLEL PSEUDONOISE SEQUENCES GENERATOR.

COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM QPSA )  
 GEORGES ROGER L.D.H. 9/11/66

1) DEGREE OF THE CHARACTERISTIC POLYNOMIAL  $(1 < p < 13)?$  7  
DEGRE DU POLYNOME CARACTERISTIQUE: 7

### INPUT OF THE CHARACTERISTIC POLYNOMIAL.

POLYNOMIALS ARE WRITTEN AS:

$$XG + A_1 X_1 + A_2 X_2 + \dots + A_{P-1} X_{P-1} + X_P$$

PLEASE GIVE THE RANK OF COEFFICIENTS A1 TO AP-1 EQUAL TO 1, ONE AFTER THE OTHER

(THOSE OF DEGREE 0 AND P ARE EQUAL TO 1 ALREADY)

INPUT 0 TO INDICATE THE END OF THE OPERATION.

RANK OF A COEFFICIENT EQUAL TO 1 ? 6

RANK OF A COEFFICIENT EQUAL TO 170

CHARACTERISTIC POLYNOMIAL: : 20 26 27

```

X? : (RETURN=YES, IF NOT, INPUT: N ) Y

```

PERIODE TROUVEE POUR SEQL: 127 PERIODE MAX: 127

00111000010111110010101100110100010011000001000000011111 : 1B3  
0011100001011111001010110001001110001010000100000000000000 : 1B2  
0011100001011111001010110001001110001010000100000000000000 : 1B1

17 : YW 30011001100 : 1A7  
18 : YW 30011001100 : 1A6  
19 : YW 30011001100 : 1A5  
20 : YW 30011001100 : 1A4  
21 : YW 30011001100 : 1A3  
22 : YW 30011001100 : 1A2  
23 : YW 30011001100 : 1A1

NUMBER OF SIMULTANEOUS BITS (HDP-1)? 6

OK? : (RETURN=YES, IF NOT, INPUT: " ) Y

DEGREE OF CHARACTERISTIC POLYNOMIAL : 7

CHARACTERISTIC POLYNOMIAL : 20 26 27

OF SIMULTANEOUS BITS: 0

COMPUTING .....

```
001110000101111001010110010100010100000100000001111111:009
001110000101111001010110010100010100000100000001111111:008
001110000101111001010110010100010100000100000001111111:007
001110000101111001010110010100010100000100000001111111:006
001110000101111001010110010100010100000100000001111111:005
001110000101111001010110010100010100000100000001111111:004
001110000101111001010110010100010100000100000001111111:003
001110000101111001010110010100010100000100000001111111:002
001110000101111001010110010100010100000100000001111111:001
```

JOB TERMINATED, RESULTS IN OPSA.DAT

```

$ TY GPSA.OAT

PARALLEL PSEUDONOISE SEQUENCES GENERATOR.

COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM GPSA )
GEORGES ROGER L.D.H. 9/11/88

DEGREE OF CHARACTERISTIC POLYNOMIAL : 7
CHARACTERISTIC POLYNOMIAL : 20 26 27
NB OF SIMULTANEOUS BITS: 0

RM07 : 25 26
RM06 : 24 25
RM05 : 23 24
RM04 : 22 23
RM03 : 21 22
RM02 : 20 21
RM01 : 25 27
RM00 : 24 26
    
```

```

*****
DEGREE OF CHARACTERISTIC POLYNOMIAL : 7
CHARACTERISTIC POLYNOMIAL : 20 26 27
NB OF SIMULTANEOUS BITS: 0
*****
    
```

MATRIX:

	0	1	2	3	4	5	6	7
0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-

表 13

```

*****
DEGREE OF CHARACTERISTIC POLYNOMIAL : 7
CHARACTERISTIC POLYNOMIAL : 20 26 27
NB OF SIMULTANEOUS BITS: 8
*****
    
```

MATRIX:

	0	1	2	3	4	5	6	7
0	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-

VERIFICATION O.K. III

PARALLEL PSEUDONOISE SEQUENCES GENERATOR.  
COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM GPSA )  
GEORGES ROGER L.D.H. 9/11/88

```

DEGREE OF CHARACTERISTIC POLYNOMIAL : 7
CHARACTERISTIC POLYNOMIAL : 20 26 27
NB OF SIMULTANEOUS BITS: 8

RM07 : 25 26
RM06 : 24 25
RM05 : 23 24
RM04 : 22 23
RM03 : 21 22
RM02 : 20 21
RM01 : 25 27
RM00 : 24 26
    
```

.....  
 DEGREE OF CHARACTERISTIC POLYNOMIAL : 7  
 CHARACTERISTIC POLYNOMIAL :  $Z^0 Z^6 Z^7$   
 NB OF SIMULTANEOUS BITS: 16  
 .....

MATRIX:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

VERIFICATION O.K. !!!

PARALLEL PSEUDONOISE SEQUENCES GENERATOR.  
 COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM GPFA )  
 GEORGES ROGER L.D.M. 9/11/88

DEGREE OF CHARACTERISTIC POLYNOMIAL : 7  
 CHARACTERISTIC POLYNOMIAL :  $Z^0 Z^6 Z^7$   
 NB OF SIMULTANEOUS BITS: 16

RN15 : 15 16  
 RN14 : 14 15  
 RN13 : 13 14  
 RN12 : 12 13  
 RN11 : 11 12  
 RN10 : 10 11  
 RN09 : 09 10  
 RN08 : 08 09  
 RN07 : 07 08  
 RN06 : 06 07  
 RN05 : 05 06  
 RN04 : 04 05  
 RN03 : 03 04  
 RN02 : 02 03  
 RN01 : 01 02  
 RN00 : 00 01

PARALLEL PSEUDONOISE SEQUENCES GENERATOR.  
 ) COMPUTATION OF THE TRANSITION MATRIX. ( PROGRAM GPFA )  
 GEORGES ROGER L.D.M. 9/11/88

DEGREE OF CHARACTERISTIC POLYNOMIAL : 7  
 CHARACTERISTIC POLYNOMIAL :  $Z^0 Z^6 Z^7$   
 NB OF SIMULTANEOUS BITS: 24

RN23 : 23 24  
 RN22 : 22 23  
 RN21 : 21 22  
 RN20 : 20 21  
 RN19 : 19 20  
 RN18 : 18 19  
 RN17 : 17 18  
 RN16 : 16 17  
 RN15 : 15 16  
 RN14 : 14 15  
 RN13 : 13 14  
 RN12 : 12 13  
 RN11 : 11 12  
 RN10 : 10 11  
 RN09 : 09 10  
 RN08 : 08 09  
 RN07 : 07 08  
 RN06 : 06 07  
 RN05 : 05 06  
 RN04 : 04 05  
 RN03 : 03 04  
 RN02 : 02 03  
 RN01 : 01 02  
 RN00 : 00 01





前記本発明の目的及び先の説明から明らかとなった目的は効果的に得られることは明らかであり、本発明の並列疑似乱数列発生機の範囲から離れずとも上記構成及び相違において所定の変更がなされ得るので、前記説明に含まれるまたは添付の図面に示された全ての事項は説明のためのものであって制限的ではないと解釈されたい。

更に、特許請求の範囲は、本明細書に記載の並列疑似乱数列発生機の一般的な及び特定の性質の全てを包含するものとし、本発明の範囲の全ての説明は、文字通りこのなかに含まれるものとする。

4. 図面の簡単な説明

第1図は次に生成される値が多項式  $1 + X^m + X^p$  によって定義されるようにP段シフトレジスタとして接続されているD型フリップフロップの使用を組み込んである直列疑似乱数列発生機を示す図、

第2図は第5B図に示された直列PRGをエミュレートする8ビット並列PRGを示すブロック図、第3図はクロック信号を含む第2図に示された8ビット並列PRGの概略図、第4図は第1図に示された直列PRGの16ビット並列PRGの実現態様のブロック図、第5A図は、段0と段0-1との間の帰還関係を示す第5B図に示された直列疑似乱数列発生機の4個の出力を有する並列PRGの実現態様の概略図、第5B図は段P及び段P-1が段1の次の値を決定するために使用される帰還値である第1図に示されたものと同様の直列PRGの概略図、第6図は第5A図に示された並列PRGに対応する直列疑似乱数列発生機の並列実施態様の一般解のための遷移行列、第7図は多項式  $1 + X^4 + X^7$  のための出力(n)と出力(n+5)及び出力(n+7)の値との関係を示す図、並びに第8図は多項式  $1 + X^2 + X^4 + X^8$  のための出力(n)と出力(n+2)、出力(n+5)及び出力(n+9)の値との関係を示す図である。

20... 直列疑似乱数列発生機、24... 並列疑似乱数列発生機、26... ラッチ、28, 30, 30'... 排他的ORゲート。

代理人 アルカイル・エヌ・ベー  
代理人 弁理士 川口 義雄  
代理人 弁理士 中村 至  
代理人 弁理士 船山 武

図面の浄書(内容に変更なし)

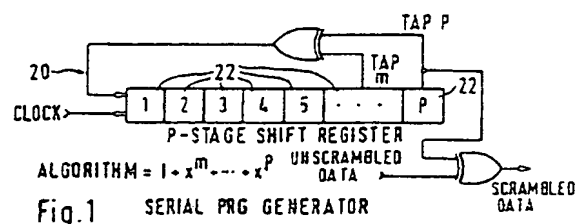
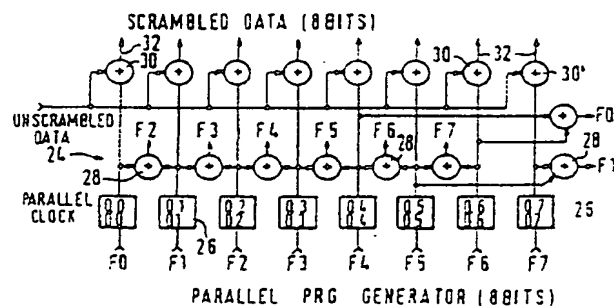


Fig. 1 SERIAL PRG GENERATOR



PARALLEL PRG GENERATOR (8BITS)

Fig. 2

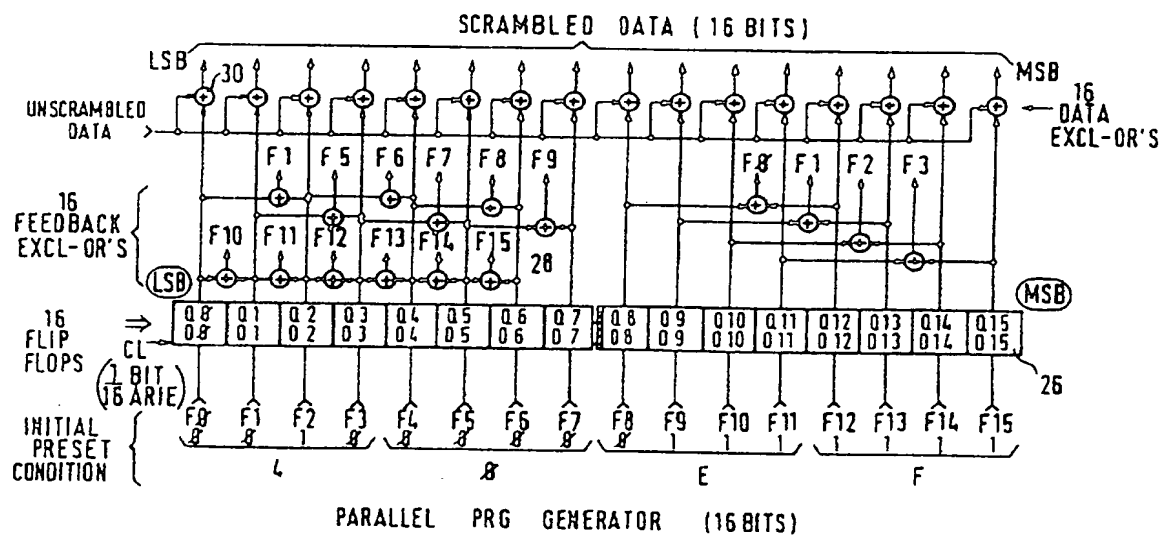
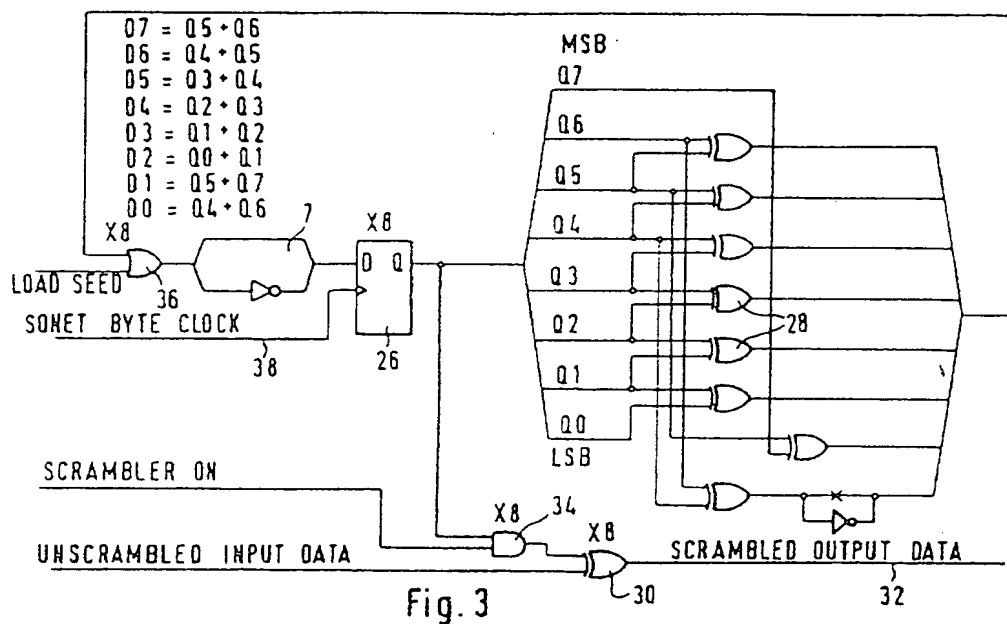
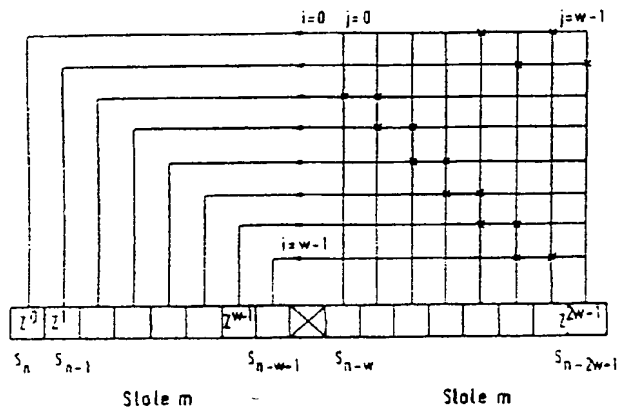
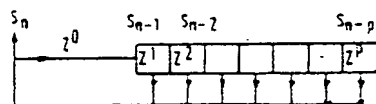


Fig. 6

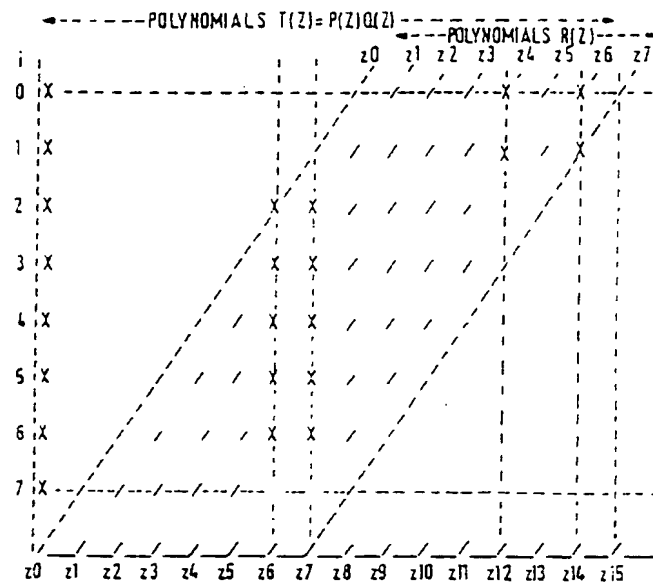


2) PARALLEL GENERATOR Fig. 5A

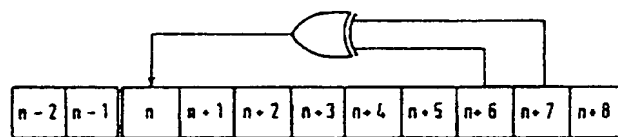


1) SHIFT REGISTER

Fig. 5B

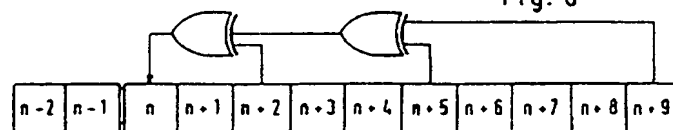


RELATIVE POSITION OF THE MATRIX ELEMENTS ('X') AND OF POLYNOMIALS  $T(z) = P(z)Q(z)$ . THE MATRIX IS IN THE PARALLELOGRAM



$$Q(n) \equiv Q(n+6) \cdot Q(n+7)$$

Fig. 7



$$Q(n) \equiv Q(n+2) \cdot Q(n+5) \cdot Q(n+9)$$

Fig. 8

第1頁の続き

②発 明 者 ジョルジュ・アンド フランス国、91240・サン・ミシエル・スユル・オルジ  
レ・シヤルル・ロジエ ユ、バルク・ドウ・ロルモイ、バビオン・マルグリット  
(番地なし)

手続補正書(方式)

平成2年9月4日

特許庁長官 樋 松 敏 殿



1. 事件の表示 平成2年特許願第122861号 /
2. 発明の名称 直列疑似乱数列生成機をエミュレートするための並列疑似乱数列生成機及びその実行方法
3. 補正をする者  
事件との関係 特許出願人  
  
名 称 アルカテル・エヌ・ベー
4. 代 理 人 東京都新宿区新宿1丁目1番14号 山田ビル  
(郵便番号160) 電話(03) 354-8623  
(5290) 弁護士 川 口 義 雄  
(ほか2名)
5. 補正指令の日付 平成2年8月28日
6. 補正の対象 図 面
7. 補正の内容  
(1) 黒色で鮮明に描いた適正な図面を別紙の通り補充する。  
(内容に変更なし)

